T · · Systems · · ·

# rvsEVO

Version 5.04

Benutzerhandbuch

rvsEVO
Version 5.04
Benutzerhandbuch


Holzhauser Straße 4 - 8
13509 Berlin

## Änderungshistorie

Folgende Änderungen wurden im Benutzerhandbuch rvsEVO durchgeführt:

### Version 5.04

- Neue Parameter für das `rvsbat`-Kommando `SEND`: INITTIME, TSTAMP, ALG, COMPRESSION, ENCRYPTION, FILEDESC, SFS, SIGN, SIGNRESP, XID
- Erweiterung der Funktionalität von `rvsbat` auf die Verwaltung von Jobstarts
- Neue Umgebungsvariablen für die Einstellung des Jobstart-Parameters **Parameterübergabe=ENV**: RVS_ERROR_TEXT und RVS_ERROR_ID
- Neue Start-/Stopp-Skripte für Installationen auf UNIX-Sytemen
- Erstellte Sendeaufträge können als Standard gesichert werden
- Revision-Dateien sind über die rvsEVO-Umgebungs-Parameter MaxRevisionLogSize und MaxRevisionLogCount konfigurierbar
- Installation mit Standardwerten - keine Dateneingabe erforderlich
- Alternative zum Stationsbaum: Anzeige in Listenform
- Neues Kapitel über Hinzufügen und Konfiguration virtueller Stationen
- Möglichkeit LogMeldungen in einer Oracle-Datenbank zu sichern
- Beschreibung der Funktionalität Connection Pooling in Kapitel 12.2 "Oracle"
- Zertifikatsverwaltung über Registerkarte **Key** erreichbar
- Neue Kommandozeilentools: `rvsbat`, `restartJob`, `convertFile`, `getCertificateList`, `sendJournal`, `startKeyMgn`
- Möglichkeit der Installation in ein vorhandenes Verzeichnis
- Verwendung von CA-Zertifikaten
- Beschreibung der Jobstartscripte im Verzeichnis $RVS_HOME\bin\jobstart
- Diverse kleinere Änderungen

### Version 5.03

- Überarbeitung des Kapitels 3.3 "Konfiguration der Jobstarts": Konfiguration von Jobstarts im Falle eines Jobverarbeitungsfehlers neue Jobstart-Parameter
- Überarbeitung des Kapitels 4.3 "Meldungen und Warnungen ausgeben": neue Möglichkeiten die Ausgabe der Log-Meldungen zu

konfigurieren nach Neustart der GUI: Anzeige der letzten 25 Log-Meldungen (konfigurierbar)

- neuer Abschnitt über die Anzeige des Lizenzschlüssels
- diverse kleinere Änderungen

**Version 5.02**

- Überarbeitung des Installations-Kapitels:

  Auswahl zwischen Server- und Client-Installation bei der Neuinstallation
  Einrichtung der Nachbarstation erst nach der Installation
  Auswahl zwischen Derby- und Oracle-Datenbank bei der rvsEVO Enterprise Edition
- neues Kapitel über die Benutzerverwaltung
- neues Kapitel über die Remote GUI
- neues Unterkapitel über Empfang ohne vorherige Entschlüsselung
- neue Komponenten für die Skripte `createSendJob` und `convertAndSend`:
  -Sd, -St, -Sh: Versand zu einem bestimmten Zeitpunkt
  -idf: Rückgabe der rvsEVO-JobID
  -xid: Integration einer externen jobID
- neue Kommandozeilentools: `getJobInfoList, deliverCertificate, requestCertificate, replaceCertificate, importCRL, terminateSession, updateStationList, holdJob, releaseJob, deleteJob`
- Erweiterung der Funktionalität Observer, um die Möglichkeit den Original-Dateinamen zu verwenden
- Restart von aktiven Sendeaufträgen im Status SP_failed
- diverse kleinere Änderungen

# 1 Einführung

Dieses Kapitel beinhaltet eine kurze Beschreibung der Produkte rvs® und rvsEVO, sowie eine Erklärung der typografischen Auszeichnungen, die in diesem Handbuch verwendet werden.

## 1.1 Kurze Beschreibung des Systems

**Was ist rvs®**

rvs® = Rechner-Verbund-System

Die Abkürzung rvs® steht für die Bezeichnung Rechner-Verbund-System. Das rvs® Rechner-Kommunikations-System ist ein etablierter Basisdienst für elektronischen Datenaustausch, EDI.

rvs® hat die Aufgabe, die Übertragung von elektronischen Daten zwischen heterogenen Computersystemen zu gewährleisten, die unterschiedliche Netzwerkprotokolle verwenden.

Um das zu erreichen, verwirklicht rvs® ein universelles Netzwerkmodell, das Sie für jeden Netzwerkknoten konfigurieren können.

rvs® sorgt für einen zuverlässigen und leistungsfähigen Transportdienst für standardisierte EDI-Nachrichten und für Dateien mit beliebigem Format und Inhalt. Sie können nur Dateien empfangen, die für rvs® vorgesehen sind. Das bedeutet, dass rvs® keinen unautorisierten Zugang zu anderen oder den eigenen Daten zulässt.

Das System wurde ursprünglich von der Volkswagen AG entwickelt und wird seit mehreren Jahren in der deutschen und europäischen Automobilindustrie genutzt, ist aber auch weltweit bei Banken, Versicherungen und in der Industrie im Einsatz.

rvs® arbeitet mit dem OFTP-Protokoll.

**Was rvs® nicht ist**

rvs® ist kein Onlinesystem. Es unterstützt weder den direkten terminalähnlichen Zugang zu anderen Rechnern, noch ermöglicht es eine Kommunikations-Pipe von Anwendung zu Anwendung auf Datensatz-Ebene. Sie können keine direkte Übertragung Ihrer Daten aus der Anwendung heraus ausführen. Sie können jedoch Sendeaufträge aus der Anwendung heraus an rvs® übergeben, die dann asynchron ausgeführt werden.

rvs® ist kein System, das Arbeitsaufgaben einplant.

rvs® interessiert sich nicht für den Inhalt der Dateien, die es überträgt. Es funktioniert als nachvollziehbares Transportmedium und führt keine Bedeutungsinterpretation der Daten durch, die es übermittelt.

rvs® ist kein EDI-Konverter. Jedoch sind zusätzliche Komponenten zur Konvertierung zwischen spezifischen Nachrichtenformaten (z.B. VDA,

ODETTE, EDIFACT, XML), die rvs® als Transportdienst benutzen, bei der T-Systems GmbH verfügbar.

rvs® ist keine Software zur Netzwerksteuerung oder Überwachung.

**Was ist rvsEVO**

rvsEVO

rvsEVO ist eine Kommunikationssoftware mit grafischer Benutzeroberfläche, die wie rvs®, auf dem OFTP-Protokoll basiert. In der näheren Zukunft ist rvsEVO als Ablösung von rvs® portable geplant, unterstützt aber zurzeit noch nicht alle Funktionalitäten von rvs® portable.

Funktionalitäten

Folgende Funktionalitäten stehen bei rvsEVO 5.4 zur Verfügung:

- grafische Benutzeroberfläche (GUI).
- Versand von Dateien an direkte Nachbar- oder an geroutete Stationen.
- Dateiempfang von direkten Nachbarstationen oder von gerouteten Stationen.
- Dateiempfang auf virtuellen Stationen; Dateiversand von virtuellen Stationen
- Aktivierung der direkten Nachbarstationen, um abholbereite Dateien zu empfangen oder die Verbindung zu testen.
- Unterstützung von OFTP Version 1.3, 1.4 oder 2.0.
- Anzeige von Informationen über aktive Empfangs- und Sendeübertragungen, über beendete und über fehlerhafte Übertragungen.
- Verfolgung von Monitoraktivitäten und Unterstützung bei der Fehleranalyse mit Hilfe der rvsEVO-Logdateien.
- Löschen oder Freigeben von Empfangsbestätigungen (EERPs).
- Codeumwandlung (ASCII -> EBCDIC und EBCDIC -> ASCII) mit verschiedenen Umwandlungstabellen.
- Formatumwandlung während der Übertragung (unterstütze Formate: Fixed, Variabel, Text, Unstrukturiert).
- Jobstarts, um die gewünschten Aktionen beim Empfang, Versand oder nach fehlerhafter Jobverarbeitung anzustoßen.
- Komprimierung und Verschlüsselung
- Backup und Recovery
- Zentrale Administration von anderen rvsEVO-Installationen.
- Unterstützung der Funktionen vom Zentralen Journal (Für mehr Informationen siehe Benutzerhandbuch Zentrales Journal)
- Unterstützung von SNMP-Monitoring (Für mehr Informationen siehe Benutzerhandbuch rvs®-SNMP-Agent).
- Zugriff auf die rvsGUI von einem entferntern Rechner (Remote GUI)

– Oracle-Datenbank-Anbindung
– PKI-Anbindung

rvsEVO ist in Java implementiert. Es existieren vier Varianten von rvsEVO: rvsEVO Enterprise Edition, rvsEVO Standard Edition, rvsEVO Light Edition und rvsEVO Tiny Edition. Den Unterschied entnehmen Sie bitte den Kapiteln 1. 2 bis 1.5.

rvsEVO kommuniziert mit anderen Applikationen über eine Batch-Schnittstelle sowie über das Dateisystem. Die angebundene Applikation kann, sofern sie in der Lage ist, die erfolgreiche Weiterverarbeitung signalisieren und den erfolgreichen Versand signalisiert bekommen.

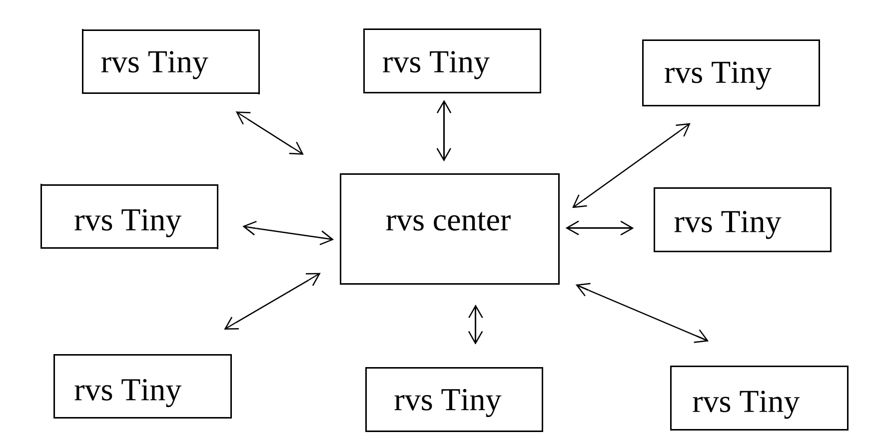
Folgende Netzwerkprotokolle werden von rvsEVO unterstützt: TCP/IP, TLS, ISDN (nur fur Windows), XOT, Proxy TCP/IP und Proxy TLS.

Nähere Informationen über unterstützte Plattformen entnehmen Sie bitte dem Dokument `$RVS_HOME\docs\liesmich.txt` (Release Notes).

**Hinweis**: Lesen Sie bitte das Kapitel 1.6 "Repräsentationsmittel" für eine Erklärung von `$RVS_HOME`.

## 1.2 rvsEVO Tiny Edition

rvsEVO Tiny Edition ist eine schlanke Variante von rvsEVO, deren Hauptmerkmal ist, dass sie nur mit einer Nachbarstation (rvs® Zentrale) direkte Netzwerkverbindung hat. Dateien, die für andere Stationen gedacht sind, werden über die rvs® - Zentrale geroutet. Daher eignet sich diese Software besonders für eine Sterntopologie, in der eine Vielzahl von Filialen oder mobilen Mitarbeitern mit einem zentralen Server verbunden ist. Folgende Abbildung soll diesen Inhalt verdeutlichen:



Die Funktionalitäten von rvsEVO Tiny Edition werden durch den Lizenzschlüssel gesteuert.

Die folgenden weiteren Merkmale sind für rvsEVO Tiny Edition kennzeichnend (Änderungen vorbehalten):

– 1 Partnerstation
– 1 Nachbarstation (rvs® - Zentrale)
– 1 Direktverbindung (Session)
– Bis zu 4 zusätzliche Routingpartner
– Kommunikationskomponente TCP/IP
– Verschlüsselung und Komprimierung
– Zentrales Journal
– SNMP-Agent
– Code-Konvertierung (PC-Mainfraime)
– zentrale Administration

## 1.3 rvsEVO Light Edition

Die rvsEVO Light Edition bietet folgendes Leistungsspektrum (Änderungen vorbehalten):

– max. 5 Partnerstationen (direkt angebundene Partner oder Routingparnter)
– keine Begrenzung der Anzahl paralleler Verbindungen (Sessions)
– Routing-Funtionalität
– Kommunikationskomponente TCP/IP
– Verschlüsselung und Komprimierung
– Derby-Datenbank
– Zentrales Journal
– SNMP-Agent
– Code-Konvertierung (PC-Mainfraime)
– zentrale Administration

Folgende Komponenten können zusätzlich eingebunden werden:

– Kommunikationskomponenten ISDN (für Windows) und XOT

## 1.4 rvsEVO Standard Edition

Die rvsEVO Standard Edition  bietet folgendes Leistungsspektrum (Änderungen vorbehalten):

- keine Begrenzung der Anzahl der Partnerstationen
- keine Begrenzung der Anzahl paralleler Verbindungen (Sessions)
- Routing-Funktionalität
- Kommunikationskomponente TCP/IP
- Verschlüsselung und Komprimierung
- Derby-Datenbank
- Zentrales Journal
- SNMP-Agent
- Code-Konvertierung (PC-Mainfraime)
- zentrale Administration
- Remote GUI (5 Benutzer)
- eine virtuelle Station

Folgende Komponenten können zusätzlich eingebunden werden:

- Kommunikationskomponenten ISDN (für Windows) und XOT
- File Service Modul

## 1.5 rvsEVO Enterprise Edition

Die rvsEVO Enterprise Edition stellt alle Grundvorraussetzungen für einen sicheren Datentransfer zur Verfügung bei einem gleichzeitigen kontinuierlichen Leistungsniveau auch bei besonders hoher Belastung.

rvsEVO Enterprise Edition  bietet folgendes Leistungsspektrum (Änderungen vorbehalten):

- keine Begrenzung der Anzahl der Partnerstationen
- keine Begrenzung der Anzahl paralleler Verbindungen (Sessions)
- Routing-Funktionalität
- alle verfügbaren Kommunikationskomponenten
- Verschlüsselung und Komprimierung
- Derby-Datenbank embedded
- Oracle-Datenbank Unterstützung
- Zentrales Journal
- SNMP-Agent
- Code-Konvertierung (PC-Mainfraime)

– zentrale Administration
– Remote GUI (10 Benutzer)
– eine virtuelle Station
– PKI-Anbindung
– File Service Modul

Folgende Komponenten können zusätzlich eingebunden werden:

– rvs® OFTP Proxy für OFTP2.

Bei Fragen zu den Editionen wenden Sie sich bitte an Ihren Vertriebspartner:

Telefon aus dem Inland: 0800 664 77 45
Telefon international: +375 606 19 902
E-Mail: rvs-service@t-systems.com

## 1.6 Repräsentationsmittel

Dieser Abschnitt enthält die Beschreibung, welche Ausprägungen und Auszeichnungen in diesem Handbuch verwendet werden und welche Bedeutung besonders gekennzeichnete Ausdrücke haben.

### Typografische Auszeichnungen

• Handlungsanweisungen beginnen mit dem Punkt als Aufzählungszeichen.

– Sonstige Aufzählungen verwenden den Halbgeviertstrich.

| | | |
|---|---|---|
| Zeichenformate | `Courier` | Kommandos, Menübefehle, Dateinamen, Pfadnamen, Programme, Beispiele, Script-Dateien, Optionen, Qualifiers, Datensätze, Felder, Modi, Fensternamen, Dialogboxen und Status |
| | **FETT** und **GROSSBUCHSTABIG** | Parameter, Umgebungsvariablen, Variablen |
| | "Hochkommata" | Verweise auf andere Handbücher, Kapitel und Abschnitte, Literatur |
| | **Fett** | wichtige Begriffe, Betriebssystemnamen, Eigennamen, Schaltflächen (Buttons), Funktionstasten. |

### Verzeichnisse

**$RVS_HOME** Weil Benutzerverzeichnisse auf unterschiedlichen Plätzen bei den unterschiedlichen Betriebssystemen zu finden sind, benutzen wir in diesem Handbuch die Variable **$RVS_HOME**. Die Standardwerte sind:

– `C:\Programme\rvsEVO` für **Windows XP** und **Windows 2000**.

Ersetzen Sie diese Variable durch Ihren richtigen Pfad.

## 1.7 Zielgruppe

Dieses Handbuch ist sowohl für Benutzer gedacht, die mit rvsEVO routinemäßig arbeiten, als auch für Administratoren. Es gibt Ihnen einen Überblick über die Basisfunktionalität von rvsEVO.

Kenntnisse Folgende Fähigkeiten sind erforderlich, um rvsEVO nutzen zu können:

– gute Kenntnisse über das benutzte Betriebssystem
– Kenntnisse über aktuell verwendete Kommunikationstechniken TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP oder Proxy TLS.

Wir empfehlen dieses Handbuch zu lesen, bevor Sie anfangen mit rvsEVO zu arbeiten.

# 2 Installation

In diesem Kapitel werden die Installationsvoraussetzungen und der Installationsvorgang von rvsEVO beschrieben.

## 2.1 Systemvoraussetzungen

Für einen erfolgreichen Betrieb von rvsEVO 5.2 benötigen Sie folgende Software:

Software

- Betriebssystem: Windows XP / Windows 2000 / Windows 2003 / Windows 7 / Windows Server 2008 / , UNIX (AIX, Solaris/SunOS, HP-UX, Linux) oder OpenVMS.
- Java Laufzeitumgebung (JRE) 1.5._XX oder Java Software Development Kit (JSDK) 1.5._XX.

  Falls auf ihrem System noch keine Java-Laufzeitumgebung installiert ist, dann holen Sie dies bitte vor der Installation von rvsEVO nach. Die Software ist frei verfügbar. Sie können diese über die Java Webseite *http://java.sun.com* herunterladen.
  **Hinweis**: auf Windows- und Linux-Systemen kann die Java-Laufzeitumgebung über den rvsEVO-Installer installiert werden.

Am Anfang brauchen Sie mindestens 200 MB freien Platz auf Ihrer Festplatte. Abhängig von der Gebrauchsintensität, der Erhaltungsdauer für alte Einträge und der Zeitspanne zwischen Datenbankbereinigungen können die Speicherplatzanforderungen erheblich größer sein.

Sie können rvsEVO über folgende Webseite herunterladen: https://servicenet.t-systems.de/tsi/de/267072/Startseite/Business-Integration/rvs

Falls Ihnen das nicht möglich ist, wenden Sie sich bitte an Ihren Vertriebspartner:

Telefon aus dem Inland: 0800 664 77 45;
Telefon international: +375 606 19 902
E-Mail: rvs-service@t-systems.com

Wir senden Ihnen die Software gerne auch auf DVD zu.

Netzwerke

rvsEVO unterstützt folgende Netzwerkprotokolle: TCP/IP, TLS, ISDN (nur Windows), XOT, Proxy TCP/IP und Proxy TLS.

**ISDN-Systemvoraussetzung**

rvsEVO ist für den Datenaustausch über das **ISDN-Netz** nur für Windows-Plattformen verfügbar.

Folgende Ausstattung wird für den Datenaustausch über das **ISDN-Netz** benötigt:

- Telefonanschluss in ISDN-Ausführung mit $s_0$-Anschluss mit mindestens zwei Nutzkanälen (B-Kanal) und einem Steuerkanal (D-Kanal)

– ISDN-Adapter
– CAPI 2.0 Treiber-Software für den Betrieb der ISDN-Karte unter Windows XP.

So installieren Sie die ISDN-Anbindung:

- Installieren Sie die ISDN-Karte in Ihren Rechner und verbinden Sie die ISDN-Karte mit dem ISDN-Anschluss.
- Stellen Sie die einwandfreie Funktion der ISDN-Karte sicher.

  **Hinweis:** Viele Kartenhersteller liefern geeignete Software für einen Selbsttest, zum Beispiel einen Anruf von einem Nutzkanal zu dem anderen Nutzkanal. Starten Sie nach der Installation und dem Test den Rechner neu, um die CAPI 2.0 Treiber-Software für andere Anwendungen verfügbar zu machen.

Externer ISDN-Router

Wenn Sie einen externen ISDN-Router mit einer "Remote CAPI" Schnittstelle (z.B. BinTec Brick) benutzen, brauchen Sie keine interne ISDN-Karte. Mehrere ISDN-Applikationen können sich dabei den gleichen Router teilen. Der Brick-Router unterstützt die "Remote CAPI" Schnittstelle. Das bedeutet, dass jeder Rechner in Ihrem LAN den Brick-Router genauso wie eine lokale ISDN-Karte benutzen kann.

**Hinweis:** T-Systems International GmbH hat folgende ISDN-Geräte erfolgreich für den Einsatz mit rvsEVO getestet:

| Gerät | Hersteller | Bemerkung |
|---|---|---|
| EICON DIVA Server BRI-2M | Eicon, http://www.eicon.de | |
| EICON DIVA Pro 2.02 PCI | Eicon, http://www.eicon.de | |
| EICON DIVA 2.0 | Eicon, http://www.eicon.de | |
| Longshine LCS-8051A | Longshine, http://www.longshine.de | |
| BIANCA/BRICK-XS, -XM, -XL2; X4000, X4100, X4300, R1200, R4100 und R4300 | Funkwerk EC, http://www.funkwerk-ec.com | |

Eine aktuelle Liste der getesteten Geräte können Sie beim rvs®-Service Support Center erhalten:

Kontakt: aus Deutschland Tel. 0800 664 77 45
aus dem Ausland Tel. +49 375 606 19 902
E-Mail: rvs-service@t-systems.com.

**XOT-Systemvoraussetzung**

Auf Anfrage liefern wir Ihnen ein separates Dokument über Einrichtung der XOT-Router. XOT bedeutet „X.25 over TCP/IP“.

Um die XOT-Funktionalität in rvsEVO zu verwenden, benötigen Sie eine TCP/IP-Verbindung zu einem XOT-Router, wie z.B. CISCO 801, CISCO 2600 und C2800 oder BINTEC X4300, R1200, R4100 und R4300 (CISCO: http://www.cisco.de; BINTEC: http://www.funkwerk-ec.com).

## 2.2 Erhalt einer Lizenz

Zum Funktionieren benötigt rvsEVO einen Lizenzschlüssel.

rvs®-Kundendienst

Um einen Lizenzschlüssel zu erhalten, wenden Sie sich bitte an das
rvs® - Service Support Center
Kontakt: aus Deutschland Tel. 0800 664 77 45
aus dem Ausland Tel. +49 375 606 19 902
E-Mail: rvs-service@t-systems.com.

Folgende Schritte sind notwendig, um einen Lizenzschlüssel zu erwerben:

– Bitte teilen Sie dem rvs®-Kundendienst den Hostnamen und die **Odette**-ID mit. Den Hostnamen erhalten Sie, indem Sie im Eingabeaufforderungsfenster (`Ausführen -> cmd`) den Befehl `hostname` eingeben. Die Odette-ID können Sie beim VDA (`http://www.vda.de/de/verband/fachabteilungen/logistik/infos/odette-id/index.html`) beantragen.
– Ihre Lizenzschlüsseldatei erhalten Sie per E-Mail.
– Die Lizenzschlüsseldatei muss im Ordner `$RVS_HOME\conf\` unter dem Dateinamen `license.properties` gespeichert werden.

Die Komponenten für die Zentrale Administration (RCI und LCI, siehe Kapitel 13) werden extra lizenziert.

**Hinweis**: Weitere Informationen über die ODETTE-ID finden Sie in Kapitel 3.2.2 "Lokale Station konfigurieren".

Lesen Sie bitte das Kapitel 1.6 "Repräsentationsmittel" für eine Erklärung von `$RVS_HOME`.

Testlizenz

Um einen Lizenzschlüssel für eine Testlizenz zu erhalten, wenden Sie sich bitte auch an den rvs®-Kundendienst. In diesem Fall ist die Mitteilung des Hostnamens und der ODETTE-ID nicht erforderlich.

## 2.3 Neuinstallation von rvsEVO

Installationsschritte

In diesem Kapitel wird die Installation von rvsEVO beschrieben. Bevor Sie mit der Installation beginnen, vergewissern Sie sich, dass Ihre

Systemumgebung alle Voraussetzungen für eine erfolgreiche Installation erfüllt (Siehe Kapitel 2.1 "Systemvoraussetzungen").

**Hinweis:** Es besteht die Möglichkeit rvsEVO ohne die Eingabe der Pflichtparameter zu installieren. In diesem Fall können Sie sich durch Betätigen der Eingabetaste durch die Installation klicken. Es werden die vom Installer vorgegebenen Standardwerte verwendet. Mit Ausnahme der Daten des Standardbenutzers können Sie die Konfiguration zu einem späteren Zeitpunkt an Ihre besonderen Bedürfnisse anpassen.
(Nicht anwendbar bei Installationen in deutscher Sprache auf der Konsole.)

Unix-Systeme

Zuerst wird die Installation auf Windows-Systemen beschrieben. Anschließend wird kurz auf die Installation auf UNIX-Systemen eingegangen, da die Installationsschritte auf beiden Betriebssystemen gleich verlaufen.

**Installation auf Windows-Systemen**

- Starten Sie Ihr Windows-System und melden Sie sich als Windows-Benutzer mit Administrator-Rechten an.
- Starten Sie die Installationssoftware `rvsEVO_X.X_setup.exe` (wobei `X.X` der Versionsnummer von rvsEVO entspricht) per Doppelklick oder über den Windows-Befehl: `Start -> Ausführen`.
- Im ersten Dialog können Sie die Sprachversion der Installation auswählen (Deutsch oder Englisch). Mit **<OK>** kommen Sie zum nächsten Schritt der Installationsroutine.

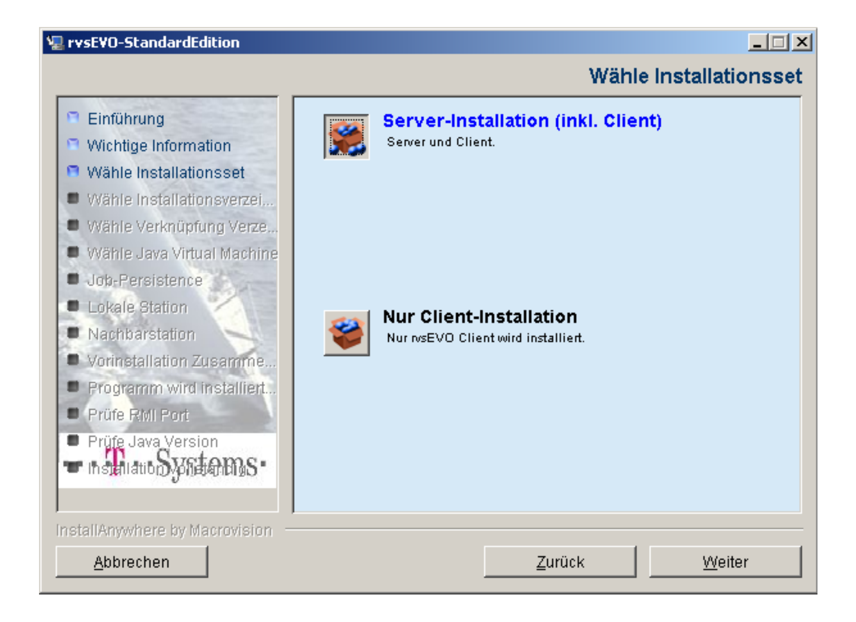



- Die beiden folgenden Fenster bieten Ihnen Informationen zur Installation und zur Version von rvsEVO.

- Anschließend wird nach bestehenden rvsEVO-Installationen gesucht. Ist bereits rvsEVO auf Ihrem Computer vorhanden, müssen Sie sich entscheiden, ob die alte Installation aktualisiert werden soll, oder ob Sie rvsEVO neu installieren möchten.

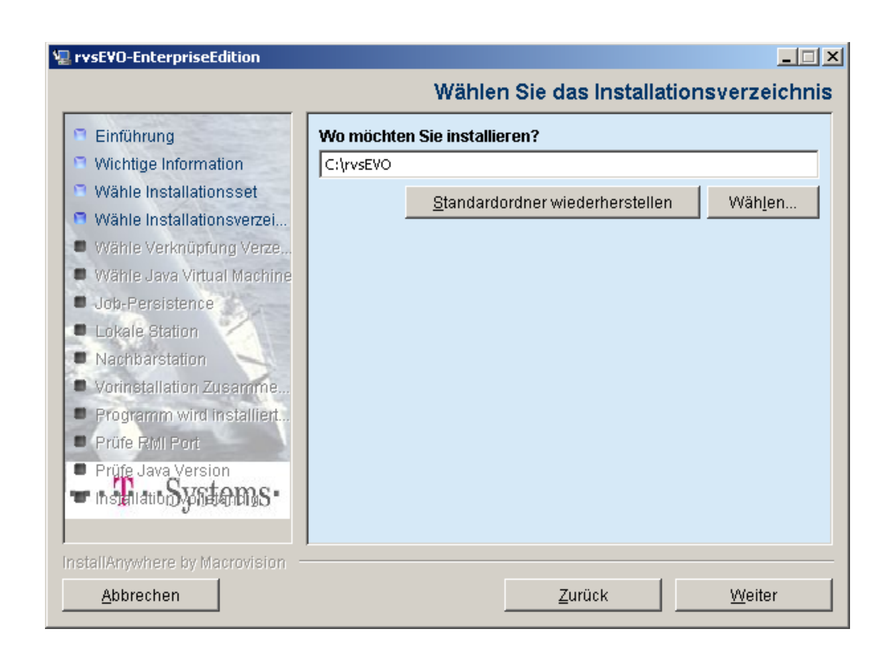



- Im folgenden Dialog können Sie zwischen einer „Server“- und einer „Client“-Installation (siehe Kapitel 9 "Remote GUI") wählen.

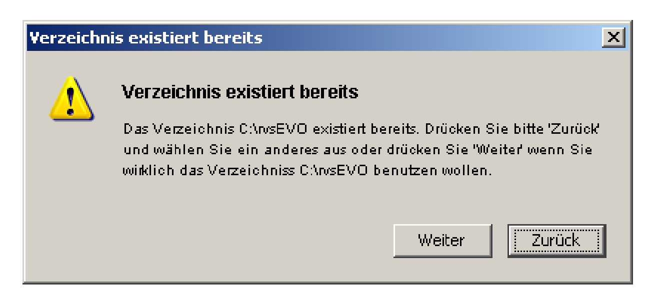



- Es folgt ein Dialog, in dem Sie das Verzeichnis angeben können, in welches rvsEVO installiert werden soll. Dieses Verzeichnis darf noch

keine rvsEVO-Installation enthalten.

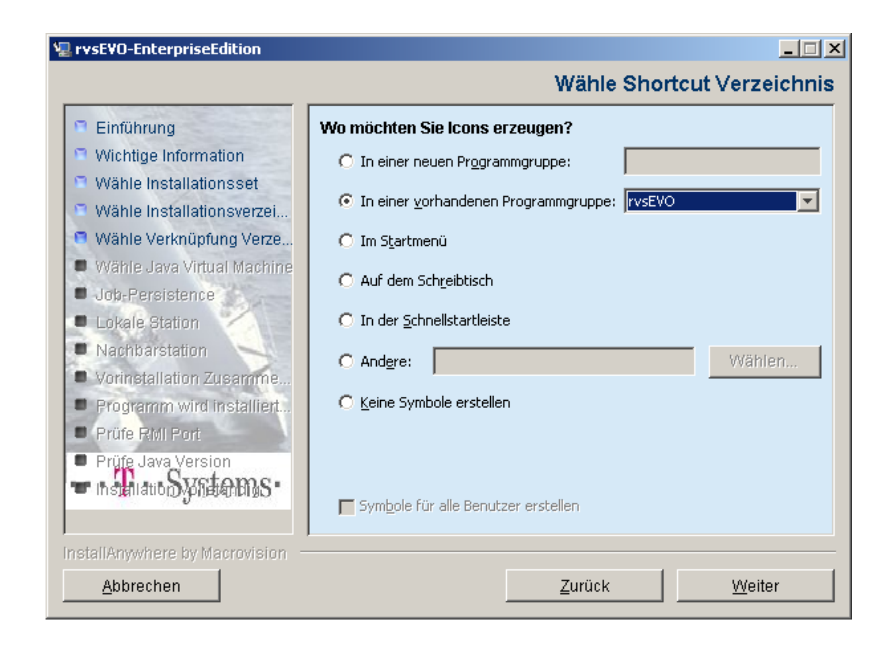



Wenn Sie ein bereits existierendes Verzeichnis auswählen, erhalten folgende Warnung:



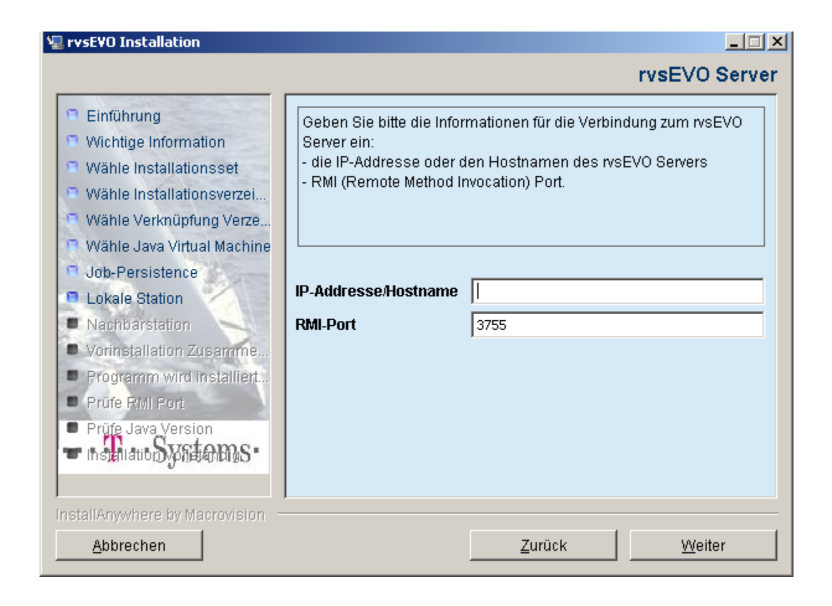

Klicken Sie auf **Weiter**, um fortzufahren.

- In folgendem Dialog legen Sie fest, in welcher Programmgruppe die Symbole für rvsEVO erstellt werden sollen.




nur Client-Installation

- Bitte folgen Sie den nächsten beiden Dialogen, wenn Sie einen rvsEVO Client installieren möchten.
  Bitte geben Sie die IP-Adresse und den RMI-Port (Standard: 3755) des rvsEVO Servers an.

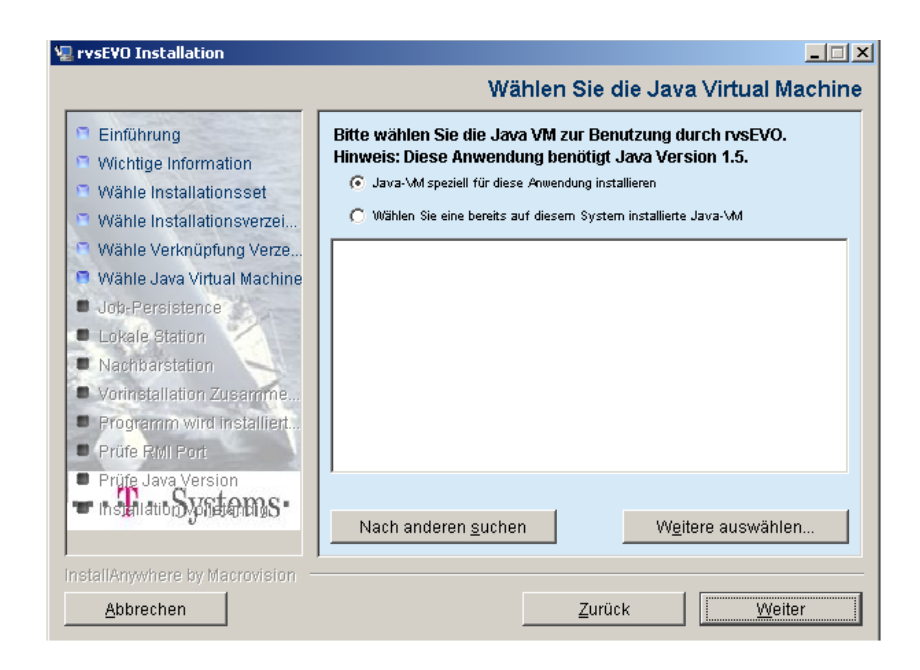



- Die letzte Bildschirmanzeige für die Client-Installation informiert Sie über die erfolgreiche Installation des rvsEVO Client.

nur Server-Installation

- Die nachfolgenden Dialoge beschreiben ausschließlich die Standard-Installation. Bitte wählen Sie aus, welche Java-Laufzeitumgebung für den Betrieb von rvsEVO eingesetzt werden soll. Sie können wählen zwischen einer von der Installationsroutine für rvsEVO mitgebrachten

JVM (Java Virtual Machine) oder einer auf Ihrem System bereits installierten JVM. Hierbei sucht das Installationsprogramm nach installierten Komponenten und schlägt Ihnen die gefundenen Versionen in einem Auswahldialog vor. rvsEVO 5.0 ist für einen Betrieb mit der Java-Version 1.5 freigegeben.




**Hinweis**: Wenn Sie sich entscheiden, eine von rvsEVO mitgebrachte JVM speziell für rvsEVO zu installieren, beantworten Sie die nächste Frage **Install JCE Files?** mit Ja (Yes).

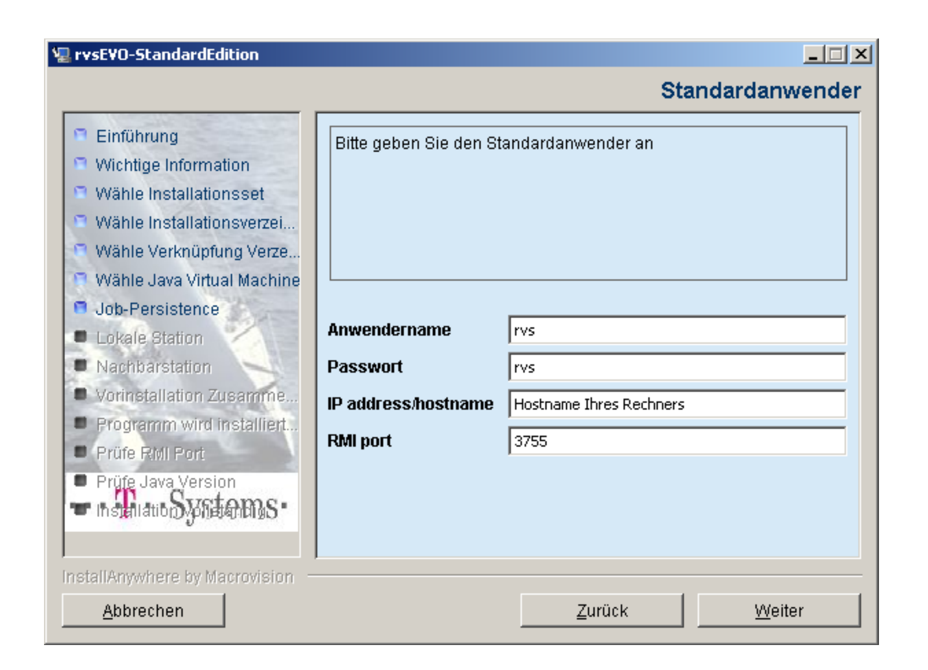
rvsEVO benutzt für die Verschlüsselung eine Erweiterung der Java Laufzeitumgebung - die JCE (Java Cryptography Extension) von Sun Microsystems, Inc. Diese wird aufgrund von Import- bzw. Exportbeschränkungen verschiedener Staaten standardmäßig in einer Variante mit Unterstützung eingeschränkter Schlüssellängen installiert. Um unbeschränkte Schlüssellängen benutzen zu können, ist es nötig, diese Erweiterung der Java-Laufzeitumgebung nachträglich zu installieren.

Im Falle, dass Sie der Installationsroutine nicht erlauben (d.h. Frage **Install JCE Files?** mit Nein beantworten) diese Komponente zu installieren, oder, dass Sie eine auf Ihrem System schon vorhandene JVM verwenden, müssen Sie die JCE-Dateien nachträglich installieren.

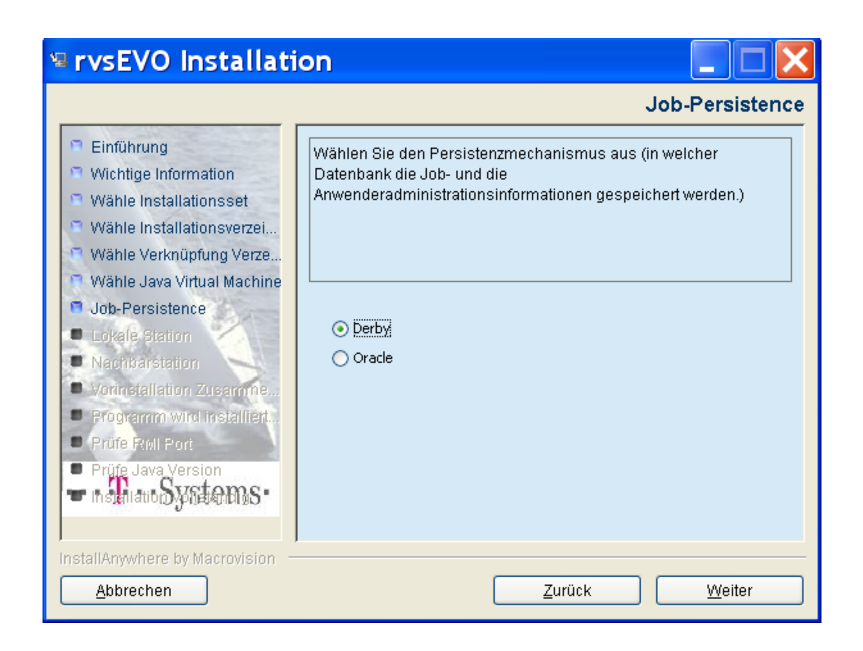
Weitere Informationen zu diesem Thema (wie die JCE-Dateien zu installieren sind oder über die JVM von IBM) finden Sie in der Datei `$RVS_HOME/docu/liesmich.txt`.

**Wichtig**: Beachten Sie, dass ohne JCE kein verschlüsselter Dateiaustausch möglich ist. Folgende Fehlermeldung erscheint, wenn Sie versuchen, die Dateien verschlüsselt auszutauschen, ohne die JCE-Dateien installiert zu haben: „Invalid key length“.

- Im nächsten Dialog legen Sie bitte den Namen und das Passwort des Standardbenutzers mit Administrator-Rechten fest. Bitte geben Sie auch den Hostnamen oder die IP-Adresse (Standard: Hostname Ihres Rechners) und den RMI-Port (Standard: 3755) an.
  **Hinweis:** Beachten Sie, dass der Default-User auf Grund seiner Sonderrolle für die lokale Client-Server-Kommunikation nicht über die Benutzerverwaltung geändert werden kann.




- Bei der Installation von rvsEVO Enterprise Edition haben Sie im nächsten Dialog die Möglichkeit, die Datenbank auszuwählen, in welcher Sie Ihre Job- und Benutzerdaten speichern möchten. Die Derby-Datenbank wird von rvsEVO mitgebracht und von der Installationsroutine im lokalen Verzeichnis `$RVS_HOME/db` installiert.




- Die nächsten drei Dialoge betreffen ausschließlich die Oracle-Datenbank. Wenn Sie sich für eine Derby-Datenbank entschieden haben, können Sie diese überspringen. In der nachfolgenden Maske werden die Oracle-Verbindungsparameter abgefragt:

– `jdbc.url` hat die folgende Syntax:

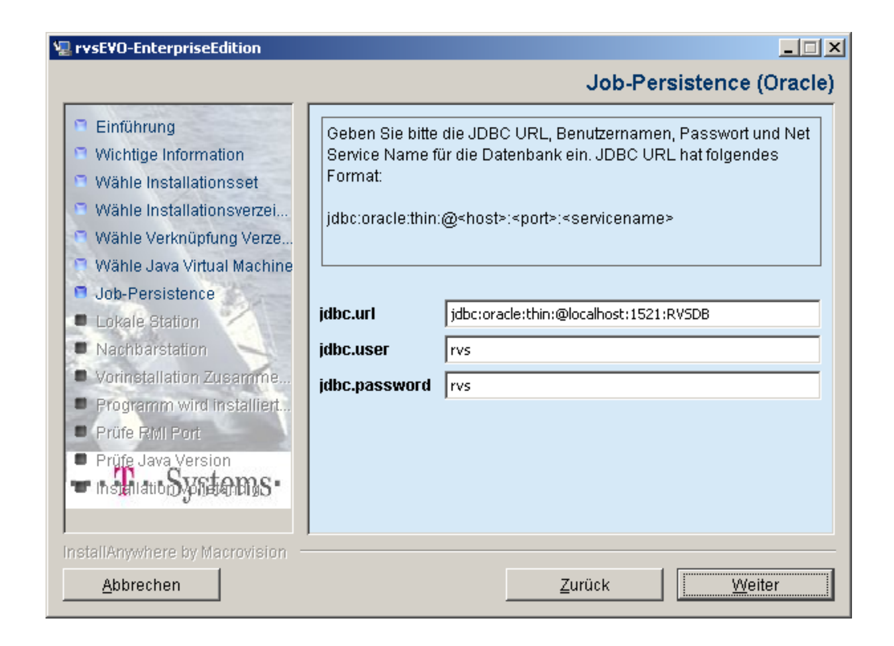
```
jdbc:oracle:thin:@<server>:<port>/<service_name>
```

`server` ist der Rechnername oder die IP-Adresse des Rechners auf dem der Oracle-Server installiert ist.
Als Standardport für Oracle ist 1521 vorgesehen.
`service_name`: Oracle-Dienstname.

– `jdbc.user` ist der Benutzer, der auf der Oracle-Datenbank eingerichtet ist

– `jdbc.passsword` ist sein Passwort.

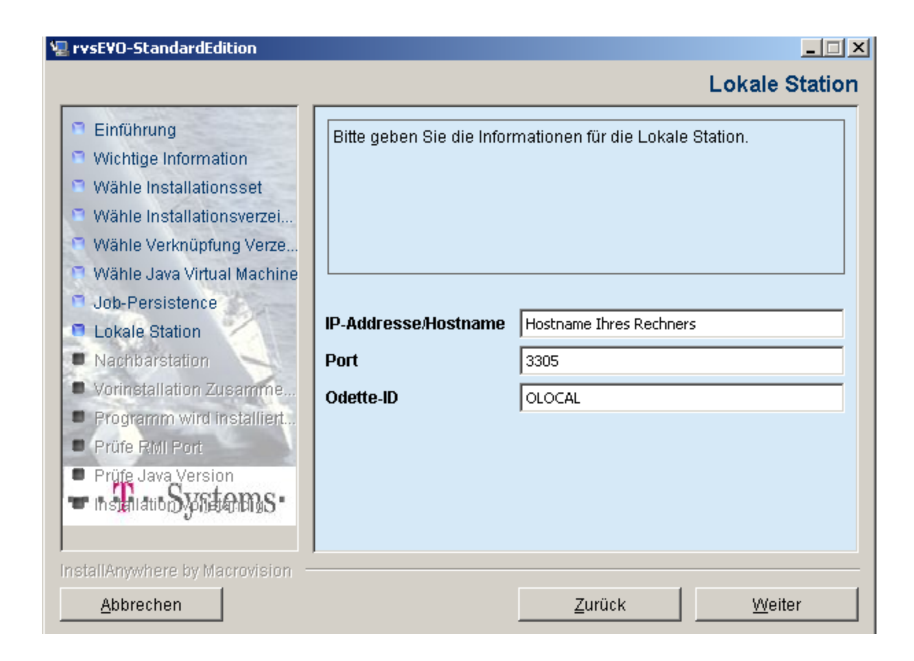
Für nähere Informationen sehen Sie bitte Kapitel 11.2 "Oracle".




- Im nächsten Fenster werden Sie gefragt, ob Ihr System nach bestehenden rvs® portable-Installationen durchsucht werden soll, damit die Einstellungen von rvs® portable importiert werden können. Wenn Sie eine Migration von rvs® portable durchführen möchten, sehen Sie bitte auch Kapitel 2.9 "Migration von rvs® portable zu rvsEVO".




- Wenn Sie die Einstellungen nicht importieren, werden Sie nachfolgend aufgefordert, die Daten der lokalen Station anzugeben. Lesen Sie bitte das Kapitel 3.2 "Anpassen der Stationskonfiguration" für ausführliche Informationen über die Stationsparameter. Eine Nachbarstation wird nach erfolgreicher Installation über die GUI eingerichtet (Siehe “Nachbarknoten hinzufügen und konfigurieren” auf Seite 64.).




- Im nächsten Dialog gibt es noch eine kurze Zusammfassung der von Ihnen festgelegten Parameter (Installationsordner, Verknüpfungsordner). Gleichzeitig erhalten Sie Informationen über den erforderlichen und tatsächlich vorhandenen Speicherplatz. Durch Betätigen der Schaltfläche **Installieren** startet die Installation und kopiert die Installationsdateien in Ihre Verzeichnisse.

– Die letzte Bildschirmanzeige informiert Sie über die erfolgreiche Installation von rvsEVO.

Unix-Systeme

## Installation auf Unix-Systemen

Wie schon am Anfang dieses Kapitels erwähnt, ist die Installation auf UNIX-Systemen analog zur Installation auf Windows-Systemen durchzuführen. Die Installationsdatei heißt `rvsEVO_X.X.X_setup.bin`

Die Installation auf UNIX-Systemen kann in verschiedenen Modi ausgeführt werden. Zur Verfügung stehen `awt`, `swing`, `console` oder `silent`.

Für die grafischen Modi (`awt` und `swing`) soll die System-Umgebungsvariable **DISPLAY** für den X-Server gesetzt werden.

**Beispiel:**

```
export DISPLAY=<IP-Adresse des X-Servers>:0.0
```

Ohne Angabe, standardmäßig, wird die Installation im Konsolenmodus ausgeführt.

**Hinweis**: Beim Aufruf der Installationsroutine, müssen Sie darauf achten, dass Sie die Installationsdatei als Shell-Skript aufrufen.

**Beispiel** (Aufruf):
`sh ./rvsEVO_300_00_SE_setup.bin -i swing`

Die Installationsabfragen in allen Modi sind gleich (Siehe Abschnitt **Installation auf Windows-Systemen**).

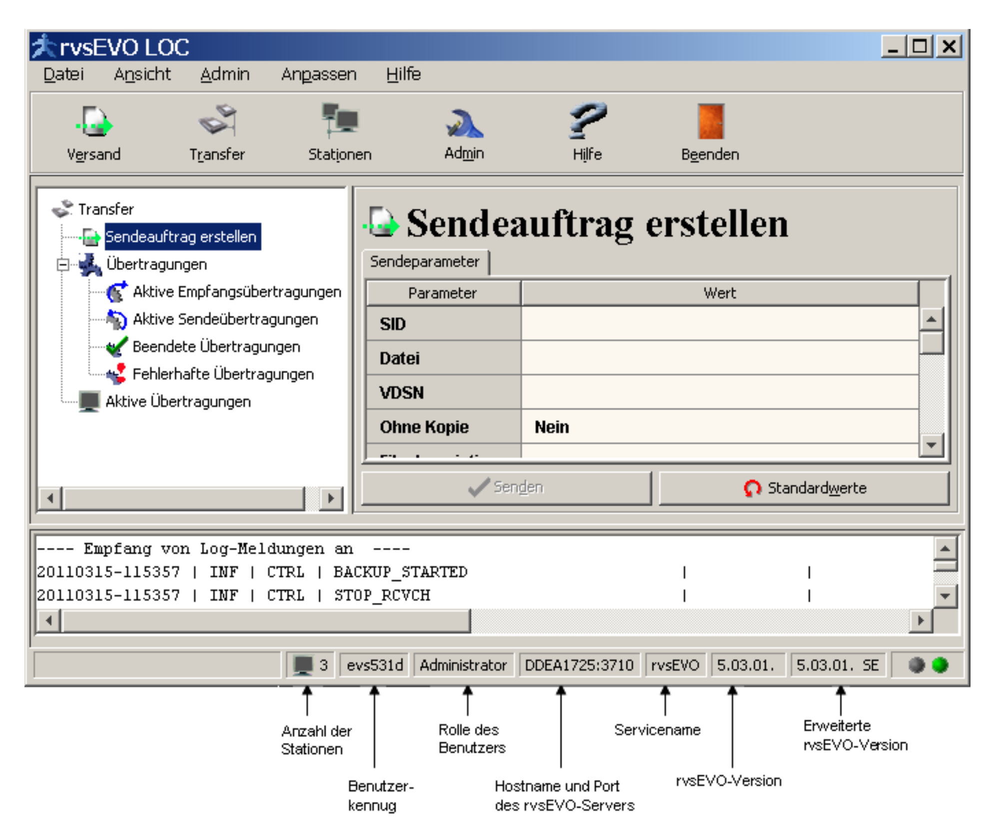
Für bestimmte UNIX-Plattformen gibt es kleine Abweichungen, die in den Release Notes für die jeweilige Version (Dokument `$RVS_HOME\docs\liesmich.txt`) beschrieben sind.

Beachten Sie bitte den Hinweis über die JCE-Dateien im Abschnitt **Installation auf Windows-Systemen**. Das Gleiche, was für Windows-Systeme gilt, gilt auch für UNIX-Systeme. Die JCE-Dateien müssen für verschlüsselten Dateiaustausch installiert sein. Mehr Informationen über dieses Thema finden Sie in der Datei `$RVS_HOME\docs\liesmich.txt`.

GUI

## 2.4 Grafische Benutzeroberfläche

Die GUI (engl.: Graphical User Interface) ermöglicht es allen Benutzergruppen, rvsEVO nach kurzer Einarbeitung zu bedienen. Die wichtigsten Funktionalitäten können über die GUI ausgeführt werden.




In der Titelleiste erscheint neben dem Programmnamen die Stations-ID der lokalen Station.

Darunter befinden sich die Menüzeile und die Funktionsleiste, in der Sie über die Symbole (Versand, Transfer, Stationen, Admin, Hilfe, Beenden) die einzelnen Menüpunkte öffnen beziehungsweise die GUI beenden können.

Der mittlere Teil ist in einen Navigationsbereich (links) und einen Arbeitsbereich gegliedert.

Im unteren Bereich des Fensters werden aktuelle Log-Meldungen angezeigt. Die Statuszeile bietet Informationen zur rvsEVO-Installation: Anzahl der Stationen, angemeldeter Benutzer, Rolle des Benutzers, IP-Adresse / Hostname und Port des rvsEVO-Servers, Servicename, rvsEVO-Version.

**Hinweis:** Die Version von rvsEVO kann ebenso mit dem Kommandozeilentool `$RVS_HOME\tools\rvsver.bat` abgefragt werden.

## 2.5 Wie starte ich rvsEVO?

**Windows**: rvsEVO starten Sie mit Hilfe der rvsEVO-Programmgruppe: `Start -> Alle Programme -> rvsEVO -> rvs GUI`.

**Unix**: Starten Sie das Shell-Skript `$RVS_HOME/bin/startGUI.sh`

Mit diesem Menüpünkt (Programm) wird zuerst die Oberfläche gestartet, die anschließend den rvsEVO-Server startet.

**Erfolgreicher Start:**

Start

Ein erfolgreicher Start sieht dann folgendermaßen aus:




`startGUI` Zum Starten der grafischen Benutzeroberfläche dient das Programm `startGUI`.

**Syntax:**

```
startGUI [-local -console -reset
-help -?]
```

Mögliche Parameter:

| | |
|---|---|
| `-local` | Start auf lokalem Rechner |
| `-console` | startet die GUI als Konsolenanwengung |
| `-reset` | setzt Größe und Position des Fensters auf die Ursprungseinstellung zurück |
| `-help` | fordert Hilfe (usage) an |
| `-?` | fordert Hilfe (usage) an |

Zum Starten des rvsEVO-Servers dient das Programm `rvsservice` oder `startServer` auf Windows-Systemen und `startServer` auf UNIX-Systemen verwendet wird. Diese Programme befinden sich im Verzeichnis `$RVS_HOME\bin\` auf Windows-Systemen in Form von Batch-Dateien und auf UNIX-Systemen in Form von Shell-Skripten.

**Hinweis für Windows**:

Auf Windows-Systemen wird rvsEVO standardmäßig als Windows-Dienst gestartet. Dies ist in der Datei `$RVS_HOME\config\rvsConfig.xml` mit dem Parameter `RvsStartScript` konfigurierbar (Siehe auch Kapitel 3.1).

Alternativ können Sie rvsEVO als Konsolenanwendung starten, indem Sie als `RvsStartScript` das Programm `$RVS_HOME\bin\startServer.bat` konfigurieren.

**Hinweis für Unix**: Auf Unix-Systemen wird standardmäßig als `RvsStartScript` das Shell-Skript `$RVS_HOME\bin\start-Server.sh` gewählt.

In den Verzeichnissen `$RVS_HOME\tools\scripts\Solaris` und `$RVS_HOME\tools\scripts\Linux` befinden sich Skripte die einen automatischen Start und Stopp des rvsEVO-Servers zusammen mit dem Betriebssystem ermöglichen. Die Readme-Dateien liefern eine Beschreibung der Installationsschritte.

## 2.6 Wie stoppe ich rvsEVO?

**Windows**: rvsEVO stoppen Sie mit Hilfe der rvsEVO-Programmgruppe: `Start -> Alle Programme -> rvsEVO -> Stop Server`.

Alternativ können Sie zum Stoppen von rvsEVO auf Windows-Systemen das Programm `$RVS_HOME\bin\stopServer.bat` aufrufen.

**Unix**: Zum Stoppen von rvsEVO wird das Shell-Skript `$RVS_HOME\bin\stopServer.sh` verwendet.

## 2.7 rvsEVO als Windows-Dienst

Windows-Dienst

rvsEVO wird standardmäßig als Windows-Dienst installiert (Siehe Kapitel 2.5).

**Hinweis**: Mit dem Begriff Dienst ist ein Programm gemeint, welches vom Betriebssystem aus gestartet werden kann und im Hintergrund arbeitet.

Die Installation von rvsEVO als Windows-Dienst ermöglicht Ihnen das Batch-Programm `$RVS_HOME\bin\rvsservice.bat`.

**Syntax:**

```
rvsservice <parameters> [options]
```

Parameter

Mögliche Parameter:

| | |
|---|---|
| `-c` | startet rvsEVO in der Eingabeaufforderung |
| `-i` | installiert rvsEVO als Windows-Dienst |
| `-r` | deinstalliert rvsEVO als Windows-Dienst |

| | |
|---|---|
| `-s` | startet den Windows-Dienst rvsEVO |
| `-h` | Hilfe |

**Beispiel:**

```
C:\Programme\rvsEVO\bin>rvsservice -i
---------------------------
-- RVSTINY SERVICE LAUNCHER --
---------------------------
rvs Server installed.
```

Mit dem Befehl `rvsservice -c` starten Sie den rvsEVO-Server in der Kommanozeile (Eingabeaufforderung).

Mit dem Befehl `rvsservice -i` installieren Sie rvsEVO als Windows-Dienst.

Startart

Nach der Installation von rvsEVO als Dienst können Sie rvsEVO in der Liste der Windows - Dienste finden (`Start -> Systemsteuerung -> Verwaltung -> Dienste`). Wenn Sie möchten, dass rvsEVO als Dienst bei jedem Systemstart automatisch gestartet wird, können Sie die Startart auf `Automatisch` setzen, indem Sie auf die Schaltfläche **Startart..**. klicken und die Startart `Automatisch` wählen.

**Beispiel**:




## 2.8 Update-Installation von rvsEVO

Update-Installation

Der Ablauf einer Update-Installation von rvsEVO ist fast identisch mit dem Ablauf einer normalen Installation (siehe Kapitel 2.3 "Neuinstallation von rvsEVO").

Voraussetzung

Bitte überprüfen Sie vor dem Start der Update-Installation, ob die Systemvariable **RVS_HOME** für den Benutzer, der die Installation durchführt, gesetzt ist (auf das Installationsverzeichnis von rvsEVO weist), da das Installationsprogramm andernfalls Ihre vorhandene rvsEVO-Installation nicht finden kann. Die Variable können Sie unter `Start -> Systemsteuerung -> System -> Erweitert -> Umgebungsvariablen` neu setzen bzw. bearbeiten.

**Wichtig:** Während des Updates wird eine neue Datei `$RVS_HOME/conf/rvs-system.properties` mit dem Namen `rvs-system.properties.new` angelegt, welche gegebenenfalls weitere Parameter enthält. Um diese neuen Funktionen nutzen zu können, ist es erforderlich, die Datei in `rvs-system.properties` umzubenennen. Bitte sichern Sie zuvor Ihre alte properties-Datei und übertragen eventuell vorgenommene Konfigurationen in die neue Datei.

Lesen Sie bitte das Kapitel 13, um zu erfahren, wie Sie mit Hilfe der Zentralen Administration andere rvsEVO-Installationen updaten können.

## 2.9 Migration von rvs® portable zu rvsEVO

Voraussetzung

Eine Migration von rvs® portable zu rvsEVO kann nur erfolgen, wenn ein lauffähiges rvs® portable auf Ihrem Rechner installiert ist und die Systemvariable **RVSENV** für den angemeldeten Benutzer gesetzt ist. Diese Variable legt fest, in welcher Datei die rvs® portable-Umgebungsvariablen definiert sind. Sie können **RVSENV** unter `Start -> Systemsteuerung -> System -> Erweitert -> Umgebungsvariablen` neu setzen bzw. bearbeiten. Die Migration sollte erst dann gestartet werden, wenn keine Kommunikation mehr stattfindet (kein Empfang/Senden der Dateien und keine Verschlüsselung/Komprimierung).

Folgende Schritte sind zu verrichten, wenn Sie eine Migration von rvs® portable zu rvsEVO durchführen möchten:

- Installieren Sie rvsEVO auf dem selben Rechner auf dem Ihr rvsXP / rvsX installiert ist unter Beachtung der folgenden Punkte:

  Beantworten Sie die Frage, ob Ihr System nach bestehenden rvs® portable-Installationen durchsucht werden soll, mit ja.




  Beantworten Sie die Frage, nach dem Import der Einstellungen mit ja, um Stationsliste und JS- / RE Einträge aus der rvs® portable-Datenbank zu importieren.




- Bei verschlüsselter Übertragung: importieren Sie Ihr eigenes ComSecure Schlüsselpaar mit dem Programm `importComSecureKeyPair` (s.u.)

- Bei verschlüsselter Übertragung: importieren Sie die öffentlichen Schlüssel Ihrer Partner, wie im Kapitel 6.5 "ComSecure-Schlüssel importieren und exportieren" beschrieben.
- Legen Sie die in rvs® portable definierten Benutzer neu an, da diese nicht übernommen werden.

`importComSecureKeyPair`

Mit dem Programm `$RVS_HOME\tools\csi\importComSecureKeyPair` können Sie im Falle einer Migration von rvs® portable zu rvsEVO ein vorhandenes ComSecure-Schlüsselpaar von rvsXP / rvsX importieren.

**Syntax:**

```
importComSecureKeyPair -k <keystore1> [-k <keystore2>]
-pri <private key> -pub <public key> -x509 <x509
filename> -s <sid> [-help] [-?]>
```

Benötigte Parameter:

| | |
|---|---|
| **-k <keystore>** | kompletter Pfad und Name der Schlüsselverwaltungsdatei von rvsEVO in der das Schlüsselpaar abgelegt werden soll (es können mehrere Dateien angegeben werden, indem der Parameter **-k** wiederholt eingegeben wird). |
| **-pri <private key>** | Name und Pfad des privaten Com-Secure-Schlüssels. |
| **-pub <public key>** | Name und Pfad des öffentlichen Com-Secure-Schlüssels (siehe auch Parameter **-x509**). |
| **-s <sid>** | StationsID der Station, der das Schlüsselpaar zugeordnet werden soll. |
| **-x509 <x509 filename>** | Name und Pfad des öffentlichen Schlüssels (anstelle von **-pub**, wenn das Zertificat im X.509-Format vorliegt). |

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |

**Wichtig:** Liegt der öffentliche Schlüssel nicht im Format X.509 vor, müssen vor dem Import des Schlüsselpaars, die Zusatzinformationen für den öffentlichen Schlüssel bereitgestellt werden. Diese sind in der Datei `$RVS_HOME\tools\csi\certificate-properties.xml` konfigurierbar.

**Beispiel (**Auszug aus `certificate-properties.xml`)**:**

```
- <!--  common-name  -->
  <entry key="subject.cn">rvsEVO-comsecure-j</entry>
```

```
- <!--  organisastion-unit  -->
  <entry key="subject.ou">rvs</entry>
- <!--  organisastion  -->
  <entry key="subject.o">T-Systems International GmbH</
entry>
- <!--  locality  -->
  <entry key="subject.l">Berlin</entry>
- <!--  state  -->
  <entry key="subject.st">Berlin</entry>
- <!--  country-code  -->
  <entry key="subject.c">de</entry>
- <!--  email  -->
  <entry key="subject.email">rvs-support@t-systems.com</
entry>
```

Im obigen Beispiel werden folgende Werte gesetzt:

– `common name` = rvsEVO-comsecure-j
– `organisation unit` = rvs
– `organisation` = T-Systems International GmbH
– `locality` = Berlin
– `state` = Berlin
– `contry code` = de
– `email` = rvs-support@r-systems.com

Beim Erzeugen eines Schlüsselpaars in rvsEVO werden diese Werte im Dialogfenster **Erzeuge Schlüsselpaar** (s. Kapitel 6.3 "Wie erzeuge ich ein eigenes Schlüsselpaar?") bestimmt.

**Wichtig:** Die übrigen Elemente der Datei `certificate-properties.xml` dürfen nicht editiert werden.

**Hinweis:** Die Einstellungen von rvs® portable können auch in eine bereits vorhandene rvsEVO-Installation übernommen werden. Dies wird mittels der CMD-Tools `$RVS_HOME\tools\portable2jobstart.bat` und `$RVS_HOME\tools\portable2stationlist.bat` ermöglicht.

### 2.10 Informationen des Lizenzschlüssels anzeigen

Um Informationen über Ihren Lizenzschlüssel zu erhalten, starten Sie bitte die GUI und wählen im Admin-Fenster den Menüpunkt `Lizenzschlüssel` aus. Sie erhalten die folgende Anzeige:




In folgender Tabelle finden Sie eine Beschreibung der ausgegebenen Parameter:

| **Parameter** | **Beschreibung** |
|---|---|
| **Company** | Firmenname des Schlüsselinhabers oder, wenn es sich um einen temporären Testschlüssel handelt: 'Testinstallation' |
| **Components** | alle lizenzierten Komponenten (s. Hinweis unterhalb der Tabelle) |
| **Key** | Ablaufdatum des Lizenzschlüssels im Format yyddd und 10-stelliger Hashkey der Lizenzdatei |
| **KeyBackup** | siehe **Key** |
| **Neighbours** | max. Anzahl der direkten Nachbarstationen |
| **Partners** | Anzahl der Partnerstationen (Nachbar- und Routingstationen) |
| **ProxyStations** | Anzahl der lizenzierten rvs® OFTP Proxy Stationen |
| **Routings** | max. Anzahl der Routingpartner |
| **Sessions** | max. Anzahl der parallelen Verbindungen |
| **Users** | max. Anzahl der Benutzer |

| | |
|---|---|
| **Virtuals** | Anzahl der lizenzierten virtuellen Stationen |

Komponenten

**Hinweis**: Es sind folgende Komponenten für rvsEVO verfügbar:

- **F**: File transfer
- **A**: Server der zentralen Administration
- **I**: Client der zentralen Administration
- **D**: externe Datenbank
- **E**: EDI Konverter
- **S**: Verschlüsselung
- **C**: Kompression
- **J**: Zentrales Journal
- **M**: SNMP-Traps
- **W**: administrative GUI über JMX (nur rvs® OFTP Proxy)
- **Y**: File Service Modul
- **K**: PKI-Anbindung
- **O**: OCSP-Zertifikatsprüfung
- **B**: CRL-Zertifikatsprüfung
- **R**: Remote GUI (nicht für Tiny Edition)
- **T**: Testkey

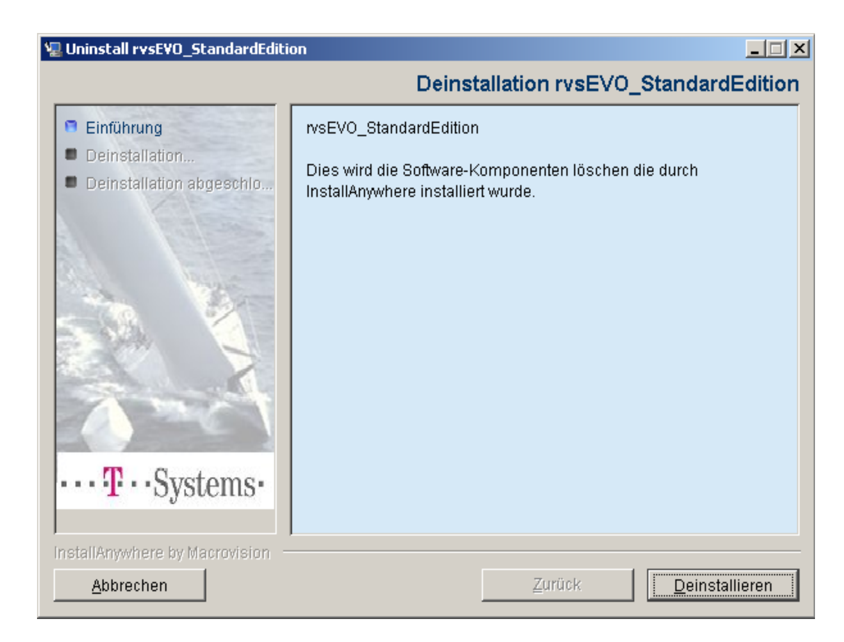
## 2.11 Deinstallation

Deinstallation

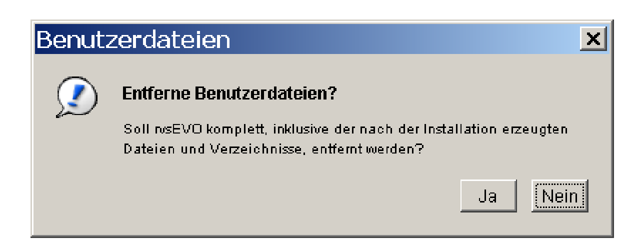
Um rvsEVO zu deinstallieren, gehen Sie bitte wie folgt vor:

- **Windows**: Starten Sie das Deinstallationsprogramm `Uninstall rvsEVO_StandardEdition.exe` im Verzeichnis `$RVS_HOME\UninstallerData` mit einem Doppelklick und folgen Sie dem Dialog.
**Unix**: Zum Starten des Deinstallationsprogramms verwenden Sie bitte das Shell-Skript `$RVS_HOME/UninstallerData/Uninstall rvsEVO_StandardEdition.sh`.
Deinstallation auf UNIX-Systemen kann analog zur rvsEVO-Installation ausgeführt werden. Folgende Modi stehen zur Verfügung: `awt`, `swing`, `silent` und `console`. Für die grafischen Modi (`awt` und `swing`) muss die System-Umgebungsvariable **DISPLAY** für den X-Server gesetzt werden (siehe Beispiel im Kapitel 2.3). Standardmäßig wird die Deinstallationsroutine im Konsolenmodus ausgeführt. Die Deinstallationsabfragen sind in allen Modi gleich.

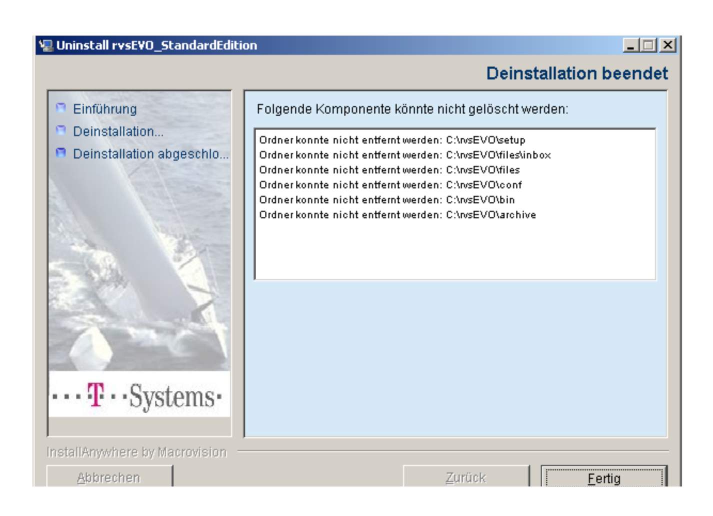
- Wählen Sie **<Deinstallieren>**, um rvsEVO von Ihrem Rechner zu entfernen.




- Im nächsten Dialog können Sie entscheiden, ob Sie auch die nach der Installation angelegeten oder geänderten Verzeichnisse und Dateien löschen möchten. Wir empfehlen dies nicht, da Sie somit alle empfangenen Dateien und Archivdateien löschen würden.´




- Das Programm wurde deinstalliert. Im folgenden Fenster wird angezeigt, welche Komponenten nicht gelöscht wurden. Klicken Sie auf **<fertig>** um das Deinstallationsprogramm zu beenden.



# 3 Konfiguration

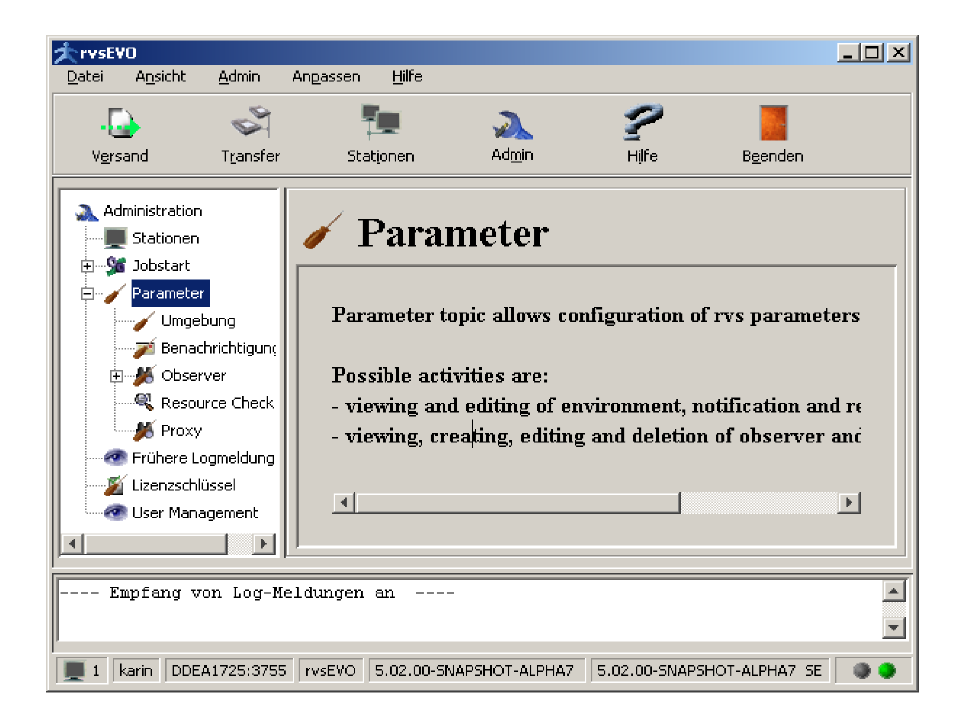
Dieses Kapitel beschreibt, wie Sie rvsEVO mittels der grafischen Benutzeroberfläche (GUI) oder mittels der XML-Konfigurationsdateien für Ihre Zwecke anpassen können.

## 3.1 Anpassen der globalen Parameter

Die **rvsEVO-Parameter** befinden sich in der Datei `rvsConfig.xml`.

Die rvsEVO-Parameter sind entweder:

– mittels der grafischen Benutzeroberfläche (`Admin -> Parameter`)




– oder mittels der XML-Konfigurationsdatei
`$RVS_HOME/conf/rvsConfig.xml`

konfigurierbar.

Es gibt die fünf folgenden Parametergruppen:

– rvsEVO-Umgebung (Environment): Dies sind allgemeine Parameter, die sich auf die rvsEVO-Umgebung beziehen.

– Benachrichtigung (SNMP): Diese Parametergruppe bezieht sich auf die Funktionalität rvs®-SNMP-Agent.

– Observer: Dies ist eine neue Funktionalität zum automatischen Durchsuchen von Verzeichnissen nach Sendeaufträgen. Sie entspricht dem Programm rvsjs in rvs® portable.

– Parameter für Speicherplatz- und Zertifikatsüberwachung: Fehlerbehandlung in Bezug auf Ressourcen-Probleme und ablaufende Zertifikate.

– Proxy: Dies sind die Parameter der Bastion Instance des rvs® OFTP Proxy. Eine Tabelle der Parameter finden Sie auf Seite 164 im Kapitel 7.3.

Diese Parametergruppen sind entweder als Untermenüpunkte des Menüpunkts Parameter im Administration-Baum sichtbar oder als separate XML-Blocks in der Datei `rvsConfig.xml` zu finden.

### 3.1.1 rvsEVO-Umgebung (Environment)

In der folgenden Tabelle sind rvsEVO-Parameter, die sich auf die rvsEVO-Umgebung beziehen, aufgeführt.

rvsEVO-Umgebungsparameter

| PARAMETER | BESCHREIBUNG |
|---|---|
| **ARCDIR** | Verzeichnis in dem Archiv- und Sicherungsdateien abgelegt werden. Standard: `$RVS_HOME\archive`. Archivdateien (RevisionLog) enthalten die Einträge der abgeschlossenen und fehlgeschlagenen Sende- und Empfangsjobs (s. Kapitel 4.12). .jar-Dateien enthalten Daten aus dem Backup und RedoLog-Dateien die dynamischen Daten ab dem Zeitpunkt der letzten Sicherung (s. Kapitel 5). |
| **BackupOnStartup** | Mit diesem Parameter kann konfiguriert werden, dass bei jedem Start von rvsEVO eine automatische Sicherung durchgeführt wird. Mögliche Werte: `Y` (Yes); `N` (No): Standardwert ist `Y`. Wenn **BackupOnStartup** auf `Y` gesetzt ist, entsteht bei jedem Start von rvsEVO im Verzeichnis `archive` eine jar-Datei. In dieser jar-Datei werden folgende rvsEVO-Verzeichnisse gesichert: `conf` und `system`. Gleichzeitig werden auch die Daten aus der Datenbank ins Verzeichnis `jobs` gesichert (siehe Kapitel 5.1 "Sicherung (Backup)" für mehr Information) . Der Name der Sicherungsdatei besteht aus dem aktuellen Datum und Uhrzeit und der laufenden dreistelligen Numerierung. Beispiel: `070104141055000.jar` ist die Sicherung, die am 04.01.07 um 14:10:55 Uhr entstanden ist. |
| **Browser** | Name des Systembrowsers, der aufgerufen wird, wenn HTML-Dateien vorliegen z.B. `explorer` |

| PARAMETER | BESCHREIBUNG |
|---|---|
| **CentralJournalInstance** | Dieser Parameter bezieht sich auf die Funktionalität „Zentrales Journal", welche in einem getrennten Handbuch beschrieben ist. **CentralJournalInstance** bedeutet die rvs®-Zielstation, an die die Journal-Dateien versendet werden. Diese Station muss in der rvsEVO-Stationsliste existieren. Lesen Sie bitte im Benutzerhandbuch „Zentrales Journal" mehr über diese Funktion. |
| **Cleanupdays** | Angabe in Tagen für die Archivierung von abgeschlossenen oder fehlgeschlagenen Jobs. Alle Jobs, die älter sind als die Zeitangabe in diesem Parameter, werden mit Hilfe des Programms `archiveJobs` (Siehe Kapitel 4.12) in die Datei `RevisionLog.xml` gespeichert (Siehe auch Parameter `PersistenceArchive`).<br>Standard: 7 (Tage)<br>**Hinweis**: Ohne Angabe von `Cleanupdays` oder `Cleanuptime` findet keine Archivierung statt. |
| **Cleanupinterval** | Zeitintervall in Minuten zwischen der Ausführung zweier Archivierungen.<br>Standard: 1440<br>**Hinweis**: Dieser Parameter ist nicht vom Parameter `Cleanuptime` oder dem Parameter `Cleanupdays` abhängig. |
| **Cleanuptime** | Angabe der Zeit im Format HHmmss für die Archivierung abgeschlossener oder fehlgeschlagener Jobs. Alle Jobs, die älter sind als die Zeitangabe in diesem Parameter, werden mit Hilfe des Programms `archiveJobs` (Siehe Kapitel 4.12) in die Datei `RevisionLog.xml` gespeichert (Siehe auch Parameter `PersistenceArchive`).<br>**Hinweis**: Ohne Angabe von `Cleanupdays` oder `Cleanuptime` findet keine Archivierung statt. |
| **ConnSetupFailWaitTime** | Zeit in Millisekunden, die rvsEVO nach dem Scheitern einer Verbindung wartet, bevor es versucht, die Verbindung erneut aufzubauen. |
| **DB** | **Hinweis**: Die Jobs werden ab der rvsEVO Version 5.0 in die Datenbank und nicht in das Verzeichnis `job` geschrieben.<br>Verzeichnis für Jobverwaltung mit den Unterverzeichnissen `ENDED`, `FAILED`, `RCV` und `SND`. In den Unterverzeichnisse `RCV` und `SND` werden die temporären, nicht vollständig abgearbeiteten Jobs gespeichert. Ins Verzeichnis `FAILED` werden die misslungenen und ins Verzeichnis `ENDED` erfolgreich abgeschlossenen Jobs abgelegt. Diese Verzeichnisse sind auch in der rvsEVO-GUI sichtbar (`Transfer -> Transmissions`). |

| PARAMETER | BESCHREIBUNG |
|---|---|
| **Description** | freier Text |
| **EngdatConfigFile** | Konfigurationsdatei für das Modul Engdat (zurzeit nicht verfügbar). |
| **FirstLanguage** | Erste Sprache für die GUI. Werte: `de` (deutsch), `en` (englisch).<br><br>Zusätzlich muss der Parameter `FirstLanguage` in der Datei `$RVS_HOME/conf/rvsEvoCli-ent.prefs` angepasst werden. In folgendem Beispiel wird deutsch als Sprache eingestellt: `<entry key="client.FirstLanguage" value="de"/>`<br>Die Änderung wird erst nach einem Neustart des rvsEVO-Servers wirksam. |
| **HelpFile** | Pfad der Hilfe-Datei. |
| **HostAllowFile** | Konfigurationsdatei, die DNS-Namen oder IP-Adressen von Hosts enthält, von denen rvsEVO Kommandos an den rvsEVO Server geschickt werden dürfen. |
| **HostDenyFile** | Konfigurationsdatei, die DNS-Namen oder IP-Adressen von Hosts enthält, von denen keine rvsEVO Kommandos an den rvsEVO-Server geschickt werden dürfen. |
| **INBOX** | Verzeichnis, in dem empfangene Dateien abgelegt werden. |
| **JobstartConfigFile** | Konfigurationsdatei für Jobstart. |
| **JournalFilenamePrefix** | Das Präfix des Journaldateinamens: Standard `TINY`. Lesen Sie bitte im Benutzerhandbuch „Zentrales Journal“ mehr über diese Funktion. |
| **LOGDIR** | Verzeichnis für die beiden Log-Dateien: `startServer.log` und `rvsservice.log`. |
| **LooptestNeighbourSID** | ID der Station, über die der Looptest (Senden einer Datei an die eigene lokale Station) ausgeführt wird. Die Datei soll an die eigene Station versendet werden und rvsEVO leitet diese Datei über eine Nachbarstation wieder an die eigene lokale. |
| **MailLocalAddress** | E-Mail-Adresse des Absenders einer Warnungs-Meldung (s. Parameter **E-Mail** und **Sende E-Mail beim Zertifikat Warnung/ Fehler** in Kaptitel 3.1.4 "Parameter für Speicherplatz- und Zertifikatsüberwachung"). |

| PARAMETER | BESCHREIBUNG |
|---|---|
| **MailSMTPHost** | Hostname oder IP-Adresse des Mailservers beim Versand von Warnungen (s. Parameter **E-Mail** und **Sende E-Mail beim Zertifikat Warnung/ Fehler** in Kaptitel 3.1.4 "Parameter für Speicherplatz- und Zertifikatsüberwachung"). |
| **ManagementConfig-File** | Name der Konfigurationsdatei für die Zentrale Administration (Siehe Kapitel über die Zentrale Administration in diesem Handbuch). |
| **MaxMonLogCount** | ohne Funktion |
| **MaxMonLogSize** | ohne Funktion |
| **MaxRevisionLog-Count** | Anzahl der Revision-Dateien `RevisionLog.xml`, die generiert werden können. Siehe Kapitel 4.12, um zu erfahren wie man eine Revision-Datei erzeugt.<br>Standard: 100 |
| **MaxRevisionLogSize** | max. Dateigröße der Revision-Datei `RevisionLog.xml`. in Zeilen (1 Zeile hat max. 10 kByte). Siehe Kap. 4.12 für weitere Informationen.<br>Standard: 1.000.000 |
| **MaxSessions** | max. Anzahl gleichzeitig laufender Empfängerprozesse für TCP/IP-Kommunikation. Standard 20; Maximum ist begrenzt durch System-Ressourcen. |
| **MonlogStylesheet** | ohne Funktion |
| **OFTPTimeout** | Time Out in Millisekunden auf der ODETTE-Ebene; Default: 30 000, kein Maximum. |
| **OUTBOX** | temporäres Verzeichnis für Versanddateien. |
| **PersistenceArchive** | Name der Datei mit den Übertragungsdaten (Statistikdaten); Standard: `$RVS_HOME/archive/RevisionLog.xml` (s. auch Kapitel 4.12). |
| **RedoLog** | legt fest, ob eine Redolog-Datei geschrieben wird. Mögliche Werte:<br>`N` (Standard) Redolog-Datei wird nicht geschrieben<br>`Y` Redolog-Datei wird geschrieben.<br>Mehr über dieses Thema können Sie im Kapitel 5 nachlesen. |
| **RMIServiceHost** | Hostname in der RMI-Registry ; Standard: localhost. RMI steht für Remote Method Invocation und ist ein Protokoll, welches für interne Prozesskommunikation in Java benutzt wird. |
| **RMIServiceName** | Name des rvsEVO-Dienstes in der RMI-Registry: Standard: rvsEVO |

| PARAMETER | BESCHREIBUNG |
|---|---|
| **RMIServicePort** | Port-Nummer für die Kommunikation in der RMI-Registry; Standard: 3755 |
| **RvsStartScript** | Aufruf des Skriptes, welches rvsEVO startet.<br>Standard:<br>`$RVS_HOME/bin/startServer.bat`<br>oder als Windows-Dienst:<br>`$RVS_HOME/bin/rvsservice.bat -s`. |
| **SendJournalInterval** | Zeitintervall in Sekunden zwischen dem Versenden zweier Journaldateien an die rvs®-Zielstation (definiert durch den Parameter `CentralJournalInstance`). Wenn hier kein Wert oder 0 angegeben wird, wird keine Journaldatei versandt. Lesen Sie bitte im Benutzerhandbuch „Zentrales Journal“ mehr über diese Funktion. |
| **SessionAliveTimeout** | Untätigkeitsdauer in Millisekunden, während der eine Verbindung noch als aktiv betrachtet wird; Standard 600 000, kein Maximum. |
| **SessionWaitTime** | Zeit in Millisekunden in der die OFTP-Session zwischen Sende- und Empfangsstation nach einer Übertragung aufrecht erhalten wird, um auf einen neuen Sendeauftrag zu warten.<br>Standard: 0 |
| **StationsConfigFile** | Stationskonfigurationsdatei: beinhaltet die Konfigurationsparameter der lokalen Station, der Nachbarstation und gerouteten Stationen. |
| **TEMP** | Verzeichnis für temporäre Nutzung. |
| **Timestamp** | legt fest, ob eine Datei bei Empfang mit einem Zeitstempel im Dateinamen abgelegt wird oder nicht.<br>Mögliche Werte:<br>`N` (Standard) Zeitstempel wird nur eingefügt, wenn der Dateiname schon existiert;<br>`Y` Dateiname wird immer mit einem Zeitstempel als Übertragungs-ID versehen. |

| PARAMETER | BESCHREIBUNG |
|---|---|
| **TraceItem** | Parameter zum Einschalten von Fehlerverfolgung (Traces).<br>Mögliche Werte:<br>`N`: keine Fehlerverfolgung (Default)<br>`O`: Fehlerverfolgung für Odette-Ebene<br>Die Ausgabe erfolgt in eine Trace-Datei, deren Name sich zusammensetzt aus SID und „_“ und SessionID und der Extension „.trc“. (z.B. `RVSEVO_6372170.trc`) Trace-Dateien werden im Verzeichnis `$RVS_HOME/log/trace/odette` gesichert. Mittels der Definition `rvs_evo.tracing.odette.station_ids=SID1,SID2,SID3` in der Datei `$RVS_HOME/conf/rvs-system.properties` können Sie die Ausgabe auf bestimmte Stationen begrenzen (hier SID1,SID2 und SID3). Zusätzlich haben Sie die Möglichkeit, die Netzwerkdaten auszugeben, indem Sie in der selben Datei die Definition `rvs_evo.tracing.odette.enable_network_data=true` aktivieren. |
| **TransmissionFail-WaitTime** | Zeit in Millisekunden für einen erneuten Start der Übertragung nach einem Misserfolg. |

### 3.1.2 Benachrichtigung (SNMP)

Die Parametergruppe Benachrichtigung bezieht sich auf die Funktionalität des rvs®-SNMP-Agenten. Das Programm **rvs®-SNMP-Agent** ist eine Anwendung, die Anfragen eines Netzwerkmanagementsystems (NMS) beantworten und Statusinformationen von rvsEVO an das NMS senden kann.

Das Installationsprogramm von **rvs®-SNMP-Agent** befindet sich in folgendem Verzeichnis: `$RVS_HOME/SNMP_Agent`. Mehr Informationen über den rvs®-SNMP-Agenten erhalten Sie im rvs®-SNMP-Agent-Benutzerhandbuch. Nach der Installation von **rvs®-SNMP-Agent** ist das Handbuch standardmäßig im Verzeichnis `C:\Programme\rvsSNMPAgent\doc` zu finden.

Folgende Parameter sind in rvsEVO zu setzen, um die Funktionalitäten von **rvs®-SNMP-Agent** nutzen zu können:

| PARAMETER | BESCHREIBUNG |
|---|---|
| **Aktiviert** | Dieser Parameter definiert, ob der rvs®-SNMP-Agent eingeschaltet ist oder nicht.<br>Standard: `Y` (Yes).<br>Mögliche Werte: `Y` (Yes) oder `N` (No). |

| PARAMETER | BESCHREIBUNG |
|---|---|
| **Intervalllänge** | Mit diesem Parameter wird definiert in welchen Abständen (in Sekunden) eine Heartbeat-Meldung von rvsEVO an die UDP-Adresse (AgentHostname + AgentPort) des Agenten gesendet wird. |
| **IP address** | Rechnername (oder IP-Adresse) des Agenten. Standard: `localhost`. |
| **LogLevel** | Dieser Parameter definiert, welche LogMeldungen von rvsEVO an den Agenten gesendet werden. Mögliche Werte: 0, 1, 2, 3.<br>`0`: keine LogMeldungen werden gesendet<br>`1`: nur Fehlermeldungen werden gesendet<br>`2`: Warnungen und Fehlermeldungen werden gesendet<br>`3`: Warnungen, Fehler- und Info-Meldungen werden gesendet |
| **Port** | IP-Port des Agenten. Standard: `3744`. |

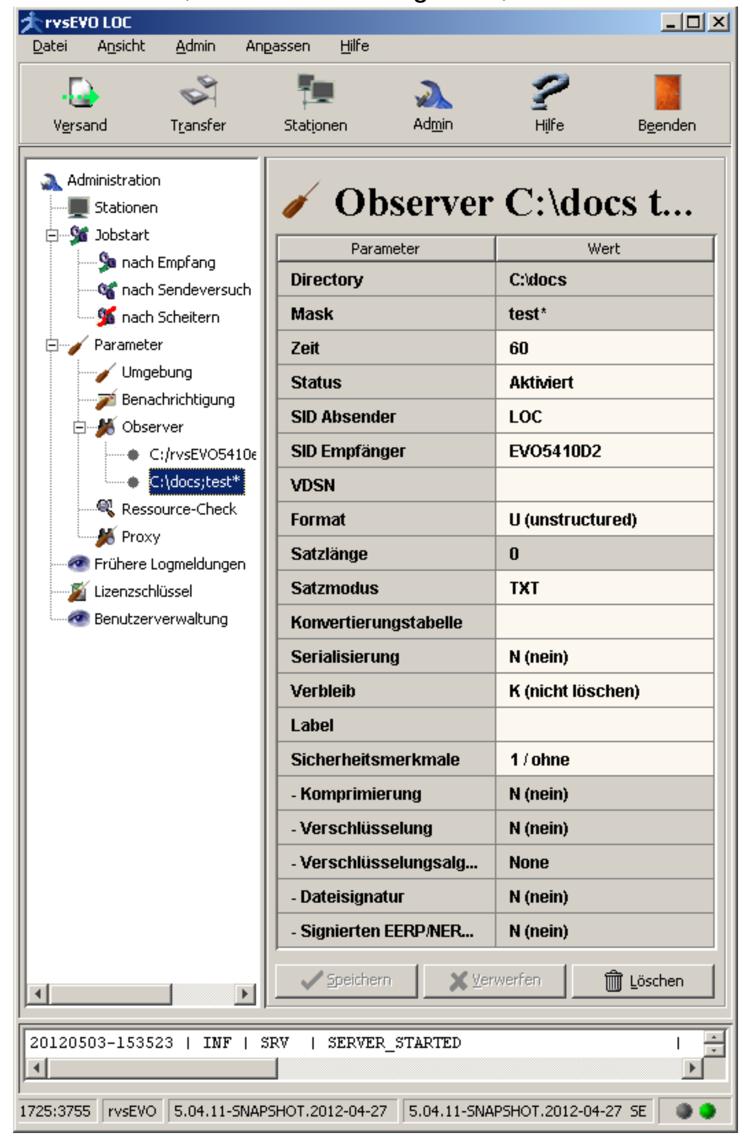
### 3.1.3 Observer

Die Funktionalität Observer dient zum Erzeugen von Sendeaufträgen für Dateien, die in ein konfigurierbares Verzeichnis gestellt werden. Der Observer untersucht in regelmäßigen Abständen das zu beobachtende Verzeichnis nach Dateinamen, die einem bestimmten Muster entsprechen. Wenn solche Dateien gefunden werden, werden aus ihnen Sendeaufträge erzeugt. Die Sendeoptionen für Sendeaufträge können konfiguriet werden. Diese Funktionalität entspricht rvsjs in rvs® portable.

Observer hinzufügen

Um einen Observer zu konfigurieren, wählen Sie im Admin-Baum den Menüpunkt **Parameter**. Nach einem rechten Mausklick auf **Observer** kann ein neuer Observer hinzugefügt werden.

**Beispiel:**
In folgendem Beispiel wird das Verzeichnis `C:\docs` alle 60 Sekunden nach Dateien, die mit `test` beginnen, untersucht.




In der folgenden Tabelle sind die Parameter, die sich auf die Funktionalität Observer beziehen, beschrieben. Diese Parameter finden Sie auch in der Konfigurationsdatei `rvsConfig.xml`, Block `Observer`)

| PARAMETER | BESCHREIBUNG |
|---|---|
| **Directory** | Das Verzeichnis, was beobachtet werden soll. |

| PARAMETER | BESCHREIBUNG |
|---|---|
| **Mask** | Die Suchmaske (regulärer Ausdruck), die angewendet werden soll z.B. `MKL*`. `MKL*` bedeutet, dass nach Dateien, die mit `MKL` beginnen, gesucht wird. Wenn eine solche Datei gefunden wird, wird für sie ein Sendeauftrag erzeugt. |
| **Status** | Dieser Parameter definiert, ob die Funktionalität Observer aktiv (eingeschaltet) ist oder nicht. Mögliche Werte: `aktiv` (eingeschaltet) und `deaktiviert` (ausgeschaltet). |
| **Zeit** | Zeitraum zwischen zwei Prüfläufen in Sekunden. |

Die restlichen Parameter werden für Versand verwendet:

`SID Absender`, `SID Empfänger`, `VDSN`, `Format`, `Satzlänge`, `Satzmodus`, `Konvertierungstabelle`, `Serialisierung`, `Verbleib`, `Label`, `Sicherheitsmerkmale`, `Komprimierung`, `Verschlüsselung`, `Verschlüsselungsalgorithmus`, `Dateisignatur` und `Signierten EERP/NERP beantragen`.

Welche Werte diese Parameter annehmen können, entnehmen Sie bitte dem Kapitel "Versand einer Datei" auf Seite 103

**Hinweis**: Es besteht die Möglichkeit den Original-Dateinamen als Dateibeschreibung zu verwenden.

Diese Einstellung ist in der Datei `$RVS_HOME/conf/rvs-system.properties` mittels der Definition `observer.sfid-desc.usefilename=true` konfigurierbar.

Observer löschen

Einen Oberser löschen können Sie, indem Sie sich den Observer in der Detailansicht anzeigen lassen (den entsprechenden Untermenüpunkt im Menüpunkt `Admin -> Parameter -> Observer` markieren) und anschließend im rechten Fensterteil die Schaltfläche **Löschen** betätigen.

### 3.1.4 Parameter für Speicherplatz- und Zertifikatsüberwachung

Diese Parametergruppe bezieht sich auf die Speicherplatzüberprüfung (Ressourcen) von rvsEVO-Verzeichnissen und Überprüfung der Ablaufdaten der Zertifikate. Mit dieser Funktionalität wird gewährleistet, dass ein rvsEVO-Operator rechtzeitig auf Speicherplatzmangel oder ablaufende Zertifikate reagieren kann.

Ressource-Check

Es werden alle aktiven Verzeichnisse von rvsEVO überprüft (Verzeichnisse, die in der Konfigurationsdatei `rvsConfig.xml` in den folgenden Variablen angegeben sind: `<DB>`, `<TEMP>`, `<INPUT>`, `<OUTPUT>`, `<ARCDIR>` und `<LOGDIR>`).

Im Falle von Speicherplatzproblemen erscheint im Monitor-Log eine Meldung, die besagt, in welchen rvsEVO-Verzeichnissen nicht ausreichend Speicherplatz vorhanden ist. Gleichzeitig kann eine E-Mail an den zuständigen Administrator gesendet werden.

Es gibt drei Stufen des Speichermangels:

– erste Stufe: es wird eine Warnung ausgegeben

– zweite Stufe: es werden alle Empfänger abgeschaltet, sodass kein Dateiempfang mehr möglich ist.

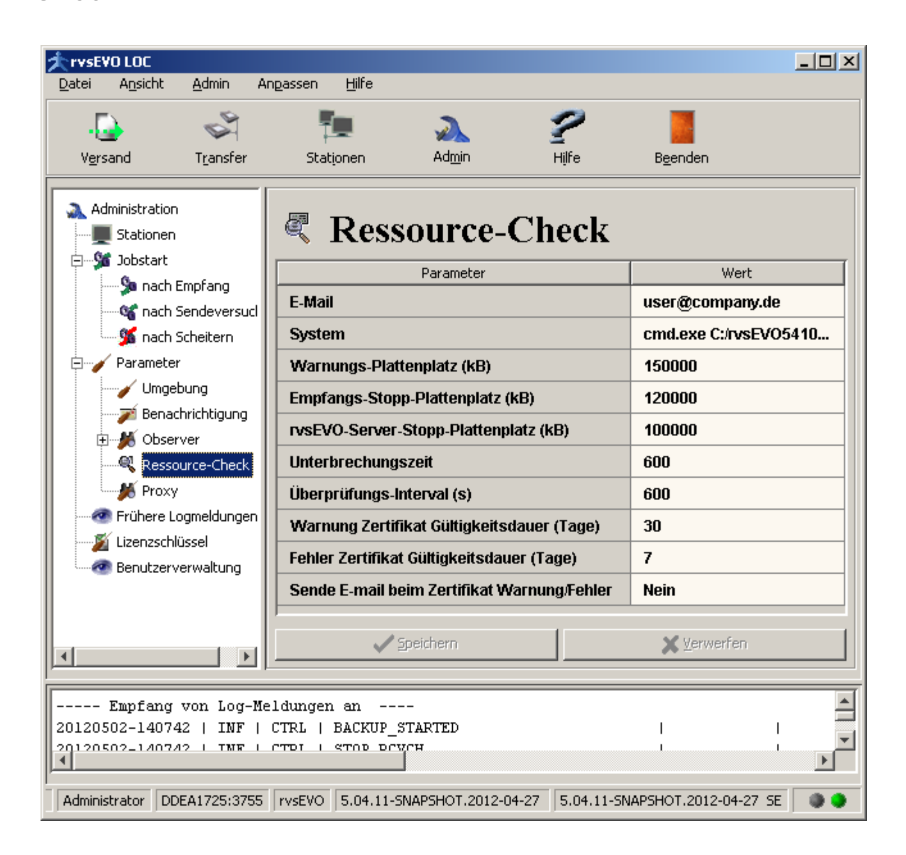
– dritte (kritische) Stufe: es wird alles abgeschaltet und rvsEVO beendet.

Für alle drei Stufen wird eine Meldung in der Log-Datei ausgegeben und es besteht gleichzeitig die Möglichkeit eine E-Mail an den zuständigen Administrator zu senden.

Zertifikats-überprüfung

Die Gültigkeitsdaten Ihrer eigenen Schlüsselpaare und der Zertifikate Ihrer Partner werden überprüft. Wenn ein Zertifikat in der vorgegebenen Zeit abläuft, wird eine Warnung/Fehlermeldung im Monitor-Log ausgegeben und bei entsprechender Einstellung eine E-Mail versendet.

Um zu dem Menüpunkt Ressource-Check zu gelangen, wählen Sie im Admin-Baum den Menüpunkt **Parameter** und anschließend **Ressource-Check**.



Folgende Parameter sind möglich:

| PARAMETER | BESCHREIBUNG |
|---|---|
| **Kritischer Festplattenplatz (CriticalDiskSpace)** | Die Anzahl von Kilobytes, die in jedem zu überprüfenden rvsEVO-Verzeichnis frei sein muss. Mit diesem Parameter soll die kritische Grenze für Speicherplatzmangel angegeben werden. Das Unterschreiten dieser Grenze bewirkt, dass alle rvsEVO-Prozesse gestoppt und rvsEVO beendet wird. Gleichzeitig haben Sie die Möglichkeit, ein Skript ausführen zu lassen (siehe Feld `System` in dieser Tabelle).<br>Standard: 100 000. |
| **E-Mail** | E-Mail-Adresse, an welche beim Server-Start eine Warnung gesendet wird, wenn die in Parameter **Festplattenplatz** angegebene Anzahl der freien Kilobytes unterschritten wird. Nach Ablauf der in Parameter **Überprüfungsintervall** festgelegten Zeit, wird erneut eine Warnung an diese Adresse versendet. S. auch Parameter **MailLocalAddress** und **MailSMTPHost** in Kapitel 3.1.1 "rvsEVO-Umgebung (Environment)" |
| **Empfangsstopfestplattenplatzt (RcvStopDiskSpace)** | Anzahl von Kilobytes, die die in jedem zu überprüfenden rvsEVO-Verzeichnis frei sein muss. Ein Unterschreiten bewirkt, dass Empfängerprozesse gestoppt werden, sodass kein Dateiempfang mehr möglich ist. Standard: 120 000 kB . |
| **Fehler Zertifikat Gültigkeitsdauer (Tage)** | Mit diesem Parameter legen Sie fest, ab welchem Tag vor Ablauf eines Zertifikates Fehlermeldungen im Monitor-Log ausgegeben werden sollen und ggf. eine E-Mail versendet werden soll (s. auch Parameter **Sende E-Mail beim Zertifikat Warnung/Fehler**).<br>Standard: 7 |
| **Festplattenplatz (DiskSpace)** | Anzahl von Kilobytes, die in jedem zu überprüfenden rvsEVO-Verzeichniss frei sein muss, bevor eine Warnmeldung in die Monitor-Log ausgegeben und ggf. eine E-Mail versendet wird (s. auch Parameter **E-Mail**).<br>Standard: 150 000 kB. |
| **Sende E-Mail beim Zertifikat Warnung/ Fehler** | E-Mail-Adresse, an welche beim Unterschreiten des in Parameter **Warnung Zertifikat Gültigkeitsdauer** sowie des in Parameter **Fehler Zertifikat Gültigkeitsdauer** festgelegten Wertes eine E-Mail versendet werden soll S. auch Parameter **MailLocalAddress** und **MailSMTPHost** in Kapitel 3.1.1 "rvsEVO-Umgebung (Environment)" |

| PARAMETER | BESCHREIBUNG |
|---|---|
| **System** | Skript, welches ausgeführt werden soll, wenn die kritische Grenze (Stufe) erreicht wurde (siehe Parameter **CriticalDiskSpace**). |
| **Unterbrechungszeit (SuspendTime)** | Zeit zwischen zwei Ressourcenprüfungen in Sekunden nach Unterschreiten des in Parameter **Empfangsstopfestplattenplatzt** angegebenen Wertes. |
| **Warnung Zertifikat Gültigkeitsdauer (Tage)** | Mit diesem Parameter legen Sie fest, ab welchem Tag vor Ablauf eines Zertifikates Warnungen im Monitor-Log ausgegeben und ggf. eine E-Mail versendet werden soll (s. auch Parameter **Sende E-Mail beim Zertifikat Warnung/Fehler**).<br>Standard: 30 |
| **Überprüfungsintervall** | Zeit in Sekunden zwischen zwei Ressourcenprüfungen<br>Standard: 600 |

**XML-Datei rvsConfig.xml**

In diesem Abschnitt geben wir ein paar Hinweise bezüglich des Aufbaus der XML-Konfigurationsdatei `rvsConfig.xml`.

Die Parameter in der Datei `rvsConfig.xml` müssen innerhalb des jeweiligen XML-Elements angegeben werden. Dabei ist die XML-Syntax zu beachten (Anfangs-Tag, Inhalt, Ende-Tag).

**Beispiel:**

```
###  Auszug aus rvsConfig.xml ###################
...
<Environment>
<TEMP>c:\Programme\rvsEVO\files\temp</TEMP>
<INBOX>c:\Programme\rvsEVO\files\inbox</INBOX>
<OUTBOX>c:\Programme\rvsEVO\files\inbox</OUTBOX>
<ARCDIR>c:\Programme\rvsEVO\archive</ARCDIR>
<JobstartConfigFile>rvsJobstart.xml</JobstartConfigFile>
<StationsConfigFile>rvsStationlist.xml</StationsConfigFile>
<PersistenceArchive>RevisionLog.xml</PersistenceArchive>
<HostAllowFile>host.allow</HostAllowFile>
<HostDenyFile> host.deny</HostDenyFile>
<RMIServiceName>rvsEVO</RMIServiceName>
<RMIServiceHost>localhost</RMIServiceHost>
</Environment>
...
########## rvsConfig.xml ###################
```

Die Änderungen in der GUI werden auch automatisch in die Datei `rvsConfig.xml` übernommen. Andersherum muss nach jeder Änderung der Datei `rvsConfig.xml` rvsEVO gestoppt und wieder gestartet werden, um Änderungen wirksam zu machen (siehe Kapitel 4.1 und 4.2).

**Hinweise**:

Sie können die Pfade für die Elemente `DB`, `TEMP`, `INBOX`, `LOGDIR` und `ARCDIR` selber festlegen und verändern.

**Beispiel:**

```
<TEMP>C:\Programme\rvsEVO\temp</TEMP>
```

Die Namen der Dateien `rvsJobstart.xml` und `rvsStationlist.xml` sind frei wählbar; sie müssen nur gültige XML-Dateien sein und sich im `conf`-Verzeichnis befinden. Das Gleiche gilt auch für Werte von Elementen `HostAllowFile` und `HostDenyFile` (siehe Parameter der 3.1.1 "rvsEVO-Umgebung (Environment)").

**Beispiel (rvsConfig.xml):**

```
<StationsConfigFile>stationen.xml</StationsConfigFile>
```

Die Datei `stationen.xml` ist eine Stationsliste im XML-Format mit den notwendigen rvsEVO-Parametern (siehe Kapitel 3.2) und befindet sich im `conf` Verzeichnis von rvsEVO.

Die Eintragungen für `RMISeviceName` und `RMIServiceHost` dienen für die interne Kommunikation und sollen nicht verändert werden.

## 3.2 Anpassen der Stationskonfiguration

Stationen

Die Konfiguration der Stationen ist auf zwei Wegen möglich:

– mittels der grafischen Benutzeroberfläche (GUI) oder

– mittels der XML-Stationskonfigurationsdatei.

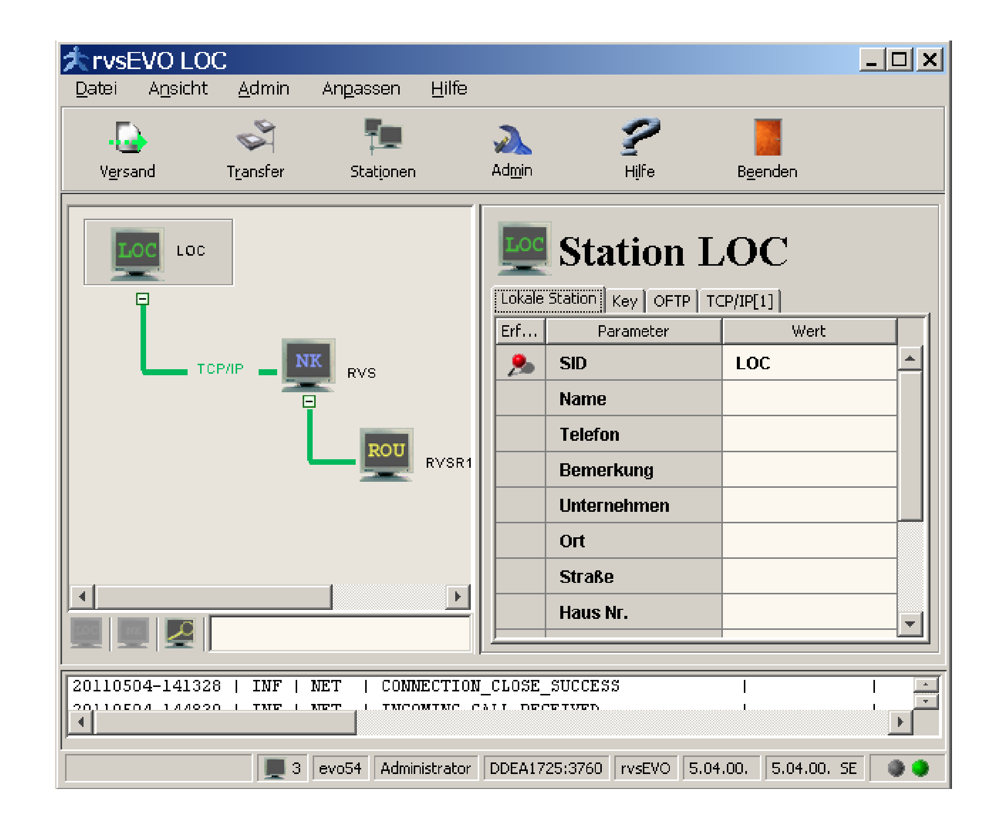
Sie benötigen einen Eintrag in der Stationskonfiguration für Ihre lokale Station, eventuellen virtuellen Stationen, die Nachbarknoten und die gerouteten Stationen, mit denen Sie die Dateien austauschen. Die obligatorischen Parameter für die lokale Station wurden schon bei der Installation abgefragt und die Stationskonfigurationsdatei

`$RVS_HOME/conf/rvsStationlist.xml`

wurde entsprechend angepasst. Diese Station ist dann auch nach dem Start der rvsEVO GUI sichtbar. Wie Sie rvsEVO starten können, lesen Sie bitte im Kapitel 2.5. Zur Einrichtung Ihrer Nachbarstation gehen Sie bitte wie im Kapitel 3.2.3 "Nachbarknoten hinzufügen und konfigurieren" beschrieben vor, die Einrichtung der gerouteten Stationen wird in Kapitel 3.2.4 "Geroutete Stationen hinzufügen und konfigurieren" beschrieben und die Einrichtung der virtuellen Stationen in Kapitel 3.2.5 "Virtuelle Stationen hinzufügen und konfigurieren".

### 3.2.1 Konfiguration von Stationen mittels GUI

Um zu den Stationen zu gelangen, wählen Sie das Symbol Stationen in der Funktionsleiste.




Standardmäßig sehen Sie im linken Bereich des Stationsfensters den Stationsbaum, im rechten Teil eine Stationsparametertabelle.

Alternativ zum Stationsbaum können Sie sich eine Liste aller Stationen ausgeben lassen. Diese Einstellung ist mittels der Definition <entry key="client.stationlist.stationtree.enabled" value="false"/> in der Datei $RVS_HOME/conf/rvsEvoClient.prefs konfigurierbar.

Der Stationsbaum/die Stationsliste stellt alle in der rvsEVO-Datenbank existierenden Stationen (Ihre lokale Station, die virtuellen Stationen, die Nachbarknoten und die gerouteten Partnerstationen) dar. Mittels Betätigung der kleinen Schaltfläche (+/-) unterhalb der lokalen Station werden die Nachbarstationen inklusive der Verbindungsart (TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP oder Proxy TLS) ein- oder ausgeblendet, ebenso die virtuellen Stationen. Der gleiche Button unterhalb einer Nachbarstation dient dem Ein- und Ausblenden der gerouteten Stationen. Per Mausklick können Sie eine der dargestellten Stationen markieren.

Registerkarten

In der Stationstabelle im rechten Fensterbereich werden Ihnen alle Parameter zur aktuell markierten Station angezeigt. Dabei können Sie

mit Hilfe der verschiedenen Registerkarten unterschiedliche Parametergruppen konfigurieren.

Graue Felder deuten darauf hin, dass diese Parameter nicht editierbar sind.

Obligatorische Parameter

Die Parameter, die für die Konfiguration einer Station obligatorisch sind, sind in der Spalte `Erf...` (Erforderlich) mit dem Symbol



gekennzeichnet.

**Beispiel:**

In der Registerkarte **OFTP** ist das `Odette Id`.

### 3.2.2 Lokale Station konfigurieren

Normalerweise werden die Parameter für das Einrichten einer lokalen Station schon bei der Installation festgelegt. In diesem Kapitel finden Sie eine ausführliche Erklärung aller für die lokalen Station möglichen Parameter.

Da die Konfiguration über die grafische Oberfläche oder mittels der XML-Konfigurationsdatei `rvsStations.xml` erfolgen kann, wird in der Parametertabelle neben dem Parameternamen auch in spitzen Klammern der Name des XML-Elements angegeben, welches den Parameter repräsentiert.

Mögliche Registerkarten für die lokale rvsEVO-Station sind: Lokale Station, Key, OFTP (Odette Parameter), TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP und Proxy TLS. Die bei der Installation nicht konfigurierten Empfänger (Netzwerke) können mit der rechten Maustaste auf die lokale Station und dem Kontextmenu hinzugefügt werden. Möglich sind TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP und Proxy TLS.

– Lokale Station: der obligatorische Parameter in dieser Registerkarte ist SID und Netzwerk. Der Parameter SID ist eine eindeutige PartnerID, die aus bis zu 16 Zeichen bestehen kann. Der Parameter Netzwerk wird automatisch angelegt, wenn Sie bei der Installation Ihre TCP/IP-Parameter eingeben. Die restlichen Parameter sind optional und dienen zur Pflege der Kontaktdaten.

– OFTP (ODETTE-Parameter): Für die lokale Station brauchen Sie nur die Odette-ID einzutragen. Die ODETTE-ID ist eine weltweit eindeutige Identifikation aller Stationen, die das ODETTE-Dateiübertragungsprotokoll (OFTP) verwenden. Der Name hat 25 Zeichen, die folgende Verteilung aufweisen:

  – der Buchstabe O,

  – eine aus 18 Zeichen bestehende Organisationsidentifikation, die von der ODETTE Kodifikationsgruppe bereitgestellt wird, und

– eine aus 6 Zeichen bestehende Unteradresse, die von jeder Organisation selbst vergeben wird.

**Hinweis**: Wenn Sie nur innerhalb Ihres geschlossenen Netzwerks kommunizieren, können Sie die Länge der ODETTE-ID frei bestimmen, so dass sie in Ihrem Netzwerk eindeutig bleibt.

Die Beschreibung der Parameter PKI und Zertifikatsvalidierung finden Sie im Abschnitt 11 "PKI-Anbindung mit rvsEVO".

– Key (Zertifikatsverwaltung): In dieser Registerkarte können Sie die Zertifikate für die Dateiverschlüsselung verwalten. Nähere Informationen finden Sie im Kapitel 6 "Verschlüsselte Übertragung mit rvsEVO (Dateiverschlüsselung)"

**TCP/IP**

In der nächsten Tabelle finden Sie die Erklärung der für das Netzwerk TCP/IP benötigten Parameter.

TCP/IP-Parameter

| Parameter | Beschreibung |
|---|---|
| **eingeschaltet <enabled>** | Dieser Parameter entscheidet, ob der TCP/IP- Empfänger gestartet werden soll, oder nicht. Mögliche Werte:<br>– `Ja` (eingeschaltet)<br>– `Nein` (ausgeschaltet) |
| **Empfänger-Nummer <ReceiverNumber>** | **Nur bei der lokalen Station**: Nummer zur Unterscheidung der verschiedenen Empfangskanäle, über die die lokale Station erreichbar ist. Zu jeder Nummer gehört eine Registerkarte mit einem Satz von TCP/IP Parametern. rvsEVO vergibt und verwaltet diese Nummer automatisch.<br>**Hinweis**: Dabei entspricht die Reihenfolge der Empfängernummern der Anordnung der Registerkarten in der grafischen Oberfläche. Die TCP/IP-Parameter in der ersten Registerkarte bekommen die Empfängernummer 1 (TCP/IP[1]); die TCP/IP-Parameter in der zweiten Registerkarte bekommen die Empfängernummer 2 (TCP/IP[2]) usw. Achten Sie beim Einrichten der Partnerstation darauf, über welche Empfängernummer die Kommunikation stattfinden soll. Siehe Parameter `Empfänger Nummer` (Partnerstation) in der Nachbarstation- Tabelle.<br>Standard: **1** |

| | |
|---|---|
| **IP-Adresse <IPAddress>** | IP-Adresse oder DNS-Name der eigenen Station. Wenn Sie bei der lokalen Station keinen Wert angeben, erlauben Sie die automatische Bestimmung der IP-Adresse. Wenn die eigene Station nur eine IP-Adresse besitzt, sollten Sie dieses Feld freilassen. |
| **max. eingehende Verbindungen <Sessions>** | Maximale Anzahl gleichzeitig laufender Empfangsprozesse über diesen Kanal.<br>Maximum: `100` |
| **Neustart Zeitintervall <RestartTimeout>** | Zeitintervall in Sekunden, nachdem ein neuer TCP/IP-Listener starten soll. |
| **Port <Port>** | Port auf dem ein TCP/IP-Listener lauschen soll; standardmäßig `3305`. |
| **Zeitüberschreitung <TimeOut>** | Abbruchzeit (time out) in Sekunden, nach der das Kommunikationsprogramm die Verbindung abbricht, wenn die Partnerstation nicht antwortet. |

**TLS**

Die TLS-Parameter werden benötigt, wenn die Kommunikation mit dem Partner verschlüsselt werden soll. TLS (Transport Layer Security) ist ein Verschlüsselungsprotokoll für Datenübertragungen im Internet. Lesen Sie bitte auch das Kapitel 3.2.6 "Konfiguration der Verschlüsselungskomponente bei einem TLS-Empfänger" um zu erfahren, wie man Schlüssel für die TLS-Kommunikation erzeugt und Zertifikate von Partnern importiert.

TLS-Parameter

| | |
|---|---|
| **Port**<br><br>**<Port>** | Port auf dem ein TLS-Listener gestartet werden soll; standardmäßig `6619` |
| **client authentication <ClientAuthentication>** | Bei einer TLS-Verbindung wird manchmal erwünscht, dass sich auch der Client (in diesem Fall, der Partner, von dem Sie die Daten empfangen), authentifiziert. Mit diesem Parameter können Sie festlegen, ob dies<br>• nicht erforderlich<br>• erwünscht oder<br>• benötigt<br>wird.<br><br>**Hinweis**: Die Authentifizierung erfolgt mittels X.509-Zertifikaten. |

| | |
|---|---|
| **eingeschaltet <enabled>** | Dieser Parameter entscheidet, ob der TLS-Empfänger gestartet werden soll, oder nicht. Mögliche Werte:<br>– `Ja` (eingeschaltet)<br>– `Nein` (ausgeschaltet) |
| **Empfänger-Nummer <ReceiverNumber>** | **Nur bei der lokalen Station**: Nummer zur Unterscheidung der verschiedenen Empfangskanäle, über die die lokale Station erreichbar ist. Zu jeder Nummer gehört eine Registerkarte mit einem Satz von TLS-Parametern. rvsEVO vergibt und verwaltet diese Nummer automatisch.<br>**Hinweis**: Dabei entspricht die Reihenfolge der Empfängernummern der Anordnung der Registerkarten in der grafischen Oberfläche. Die TLS-Parameter in der ersten TLS-Registerkarte bekommen die Empfängernummer 1(TLS[1]); die TLS-Parameter in der zweiten TLS-Registerkarte bekommen die Empfängernummer 2 (TLS[2]) usw. Achten Sie beim Einrichten der Partnerstation darauf, über welche Empfängernummer die Kommunikation stattfinden soll.<br>Siehe Parameter `Empfänger Nummer` (Partnerstation) in der Tabelle der Nachbarstation.<br>Standard: `1` |
| **IP-Adresse <IPAddress>** | IP-Adresse oder DNS-Name der eigenen Station für die TLS-Verbindung. |
| **max. eingehende Verbindungen <Sessions>** | Maximale Anzahl gleichzeitig laufender Empfangsprozesse über diesen Kanal. Maximum: 100 |
| **Neustart Zeitintervall <RestartTimeout>** | Zeitintervall in Sekunden nachdem ein neuer TLS-Empfänger (Listener) starten sollte. |
| **Parameter** | **Beschreibung** |
| **Schlüsseldateiname für vertrauenswürdige Zertifikate <TrustManagerKeystoreFileName>** | Name der Schlüsselverwaltungsdatei für vertrauenswürdige Zertifikate (trust certificates). In diese Datei werden die vertrauenswürdigen X.509-Zertifikate (öffentliche Schlüssel Ihrer Partner in Form von X.509-Zertifikaten) importiert. Wir empfehlen den Standard-Schlüsseldateiname zu nutzen: `$RVS_HOME/system/data/tlsTrustKeyManagerStore.p12`. |

| | |
|---|---|
| **Schlüsseldateiname <KeystoreFileName>** | Name der Schlüsselverwaltungsdatei für die TLS-Verbindung. In dieser Datei sind eigene privaten und öffentlichen Schlüsslel aufbewahrt. Wir empfehlen die Standard-Schlüsselverwaltungsdatei zu nutzen: `$RVS_HOME/system/data/tlsKeyManagerStore.p12`. |
| **Zeitüberschreitung <TimeOut>** | Abbruchzeit (time out) in Sekunden, nach der das Kommunikationsprogramm die Verbindung abbricht, wenn die Partnerstation nicht antwortet |

**Proxy TCP/IP und Proxy TLS:**

Die Parameter von Proxy TCP/IP und Proxy TLS sind nahezu identisch mit denen von TCP/IP und TLS. Es gibt lediglich einen zusätzlichen Parameter:

- Proxy Bastion: als Wert können Sie aus einer Liste, der in rvsEVO angelegten Bastion Instances, die Bastion auswählen, auf der der Empfänger/Listener gestartet soll.

### ISDN

Lesen Sie bitte das Kapitel 2.1, um zu erfahren, welche Systemvoraussetzungen für die ISDN Kommunikation notwendig sind. ISDN als Netzwerktyp ist nur für Windows Plattformen verfügbar.

In der folgenden Tabelle werden die ISDN-Parameter der lokalen Station beschrieben.

ISDN-Parameter

| Parameter | Beschreibung |
|---|---|
| **eingeschaltet <enabled>** | Dieser Parameter entscheidet, ob der ISDN-Empfänger gestartet werden soll, oder nicht. Mögliche Werte:<br>– `Ja` (eingeschaltet)<br>– `Nein` (ausgeschaltet). |

| | |
|---|---|
| **Empfänger-Nummer <ReceiverNumber>** | **Nur bei der lokalen Station**: Nummer zur Unterscheidung der verschiedenen Empfangskanäle, über die die lokale Station erreichbar ist. Zu jeder Nummer gehört eine Registerkarte mit einem Satz von ISDN-Parametern. rvsEVO vergibt und verwaltet diese Nummer automatisch.<br>**Hinweis**: Dabei entspricht die Reihenfolge der Empfängernummern der Anordnung der Registerkarten in der grafischen Oberfläche. Die ISDN-Parameter in der ersten ISDN-Registerkarte bekommen die Empfängernummer 1(ISDN[1]); die ISDN-Parameter in der zweiten ISDN-Registerkarte bekommen die Empfängernummer 2 (ISDN[2]) usw. Achten Sie beim Einrichten der Partnerstation darauf, über welche Empfängernummer die Kommunikation stattfinden soll. Siehe Parameter `Empfänger Nummer` (Partnerstation) in der Tabelle der Nachbarstation.<br>Standard: `1` |
| **Empfänger-Timeout <RestartTimeOut>** | **Nur gültig für die lokale Station**: Abbruchzeit in Sekunden, nach der der Empfänger das Warten auf ankommende Anrufe unterbricht und sich wieder neu bei dem Kartentreiber anmeldet. |
| **ISDN Facilities <Facilities>** | Spezielle Informationen oder Anlagen der ISDN-Übertragung. |
| **ISDN Nummer <Address>** | ISDN Nummer unter der die Station erreichbar ist. **Lokale Station**: Eigene ISDN-Nummer ist immer ohne Orts- und Länderkennung anzugeben. Bei ISDN-Standard 1TR6 wird für die lokale Station die einstellige EAZ (**E**nd**A**uswahl**Z**iffer) vergeben. |
| **ISDN Protokoll <Protocol>** | Gibt den genutzten ISDN-Standard an.<br>- `1TR6` Deutscher Landesstandard<br>- `E-DSS1` EURO-ISDN<br>Standard: `E-DSS1` |
| **ISDN Terminal Identifier <Tei>** | Nur nötig für X.31: **T**erminal **E**nd **I**dentification.<br>Standard: `0` – keine TEI vergeben |
| **ISDN Userdata <Userdata>** | Benutzerangaben für die ISDN-Übertragung |

| | |
|---|---|
| **Kartennummer <DeviceCardNumber>** | Nummer der im Rechner installierten ISDN-Karte, beginnend mit "0" |
| **max. eingehende Verbindungen <Sessions>** | Maximale Anzahl von parallel arbeitenden Verbindungen. Für den Empfang gilt: Sobald ein Anruf angenommen wurde, startet rvsEVO einen neuen Empfänger, bis die maximale Anzahl erreicht wird. |
| **Typ <DeviceType>** | nicht konfigurierbar. Kennzeichnet die Art der Kommunikation in ihrem Rechnr - CAPI2; Standard: CAPI2A (A=no Diehl/Eicon card) |
| **Verbindungs-Timeout <Timeout>** | Abbruchzeit in Sekunden, nach der das Kommunikationsprogramm die Verbindung abbricht, wenn die Partnerstation oder die CAPI nicht antworten.<br>Standard: `30` |
| **X.25 Adresse <X25Address>** | 15-stellige X.25 DTE Adresse der Station. Angabe ist optional. Es wird jedoch empfohlen, hier die ISDN Nummer einzutragen, da manche Gegenstellen eine X.25 Adresse erwarten. |
| **X.25 Dbit <DBit>** | Das D-Bit ist ein Feld im Datenpaket von X.25, das für die Ende-zu-Ende-Bestätigung benutzt wird. Die DTE zeigt damit an, ob sie den Empfang einer Ende-zu-Ende-Bestätigung wünscht.<br>Standard: `Nein` |
| **X.25 Facilities <X25Facilities>** | Spezielle Informationen oder Anlagen bei X.25-Übertragung; siehe Informationen des benutzten X.25-Dienstes. |
| **X.25 Fenstergröße <WindowSize>** | Die Fenstergröße bei X.25/ISDN-Kommunikation ist die Anzahl der Pakete, die ohne Bestätigung verschickt werden können. Die Fenstergröße kann während des Verbindungsaufbaus ausgehandelt werden. Wir empfehlen aber von vornherein die zum Partnernetzwerk passende Fenstergröße zu konfigurieren. **7** empfholen für ISDN, 2 für X.25 native. |

| | |
|---|---|
| **X.25 geschlossene Benutzergruppe (closed user group) <ClosedUserGroup>** | ISDN und X.25 bieten die Möglichkeit, eine geschlossene Benutzergruppe zu bilden. Alle Teilnehmer, die zu einer solchen Gruppe gehören, können untereinander über das öffentliche Telekommunikationsnetz kommunizieren. Verbindungsanforderungen an Gruppenmitglieder von Teilnehmern, die nicht Mitglied der geschlossenen Benutzergruppe sind, werden von der Vermittlungsstelle abgewiesen. Dies gilt gleichermaßen für Verbindungsanforderungen von Gruppenmitgliedern an Teilnehmer, die nicht Mitglied der geschlossenen Benutzergruppe sind. Dieses Dienstemerkmal wird als Closed User Group (CUG) bezeichnet.<br>Standard: `Nein` |
| **X.25 Paketgröße <PacketSize>** | Größe der Datenpakete bei der Datenübertragung.<br>Standard: `128` |
| **X.25 Userdata <X25UserData>** | Benutzerangaben für X.25-Übertragung; siehe Informationen des benutzten X.25-Dienstes |

**XOT:**

XOT Router sind in der Lage X.25-Pakete zwischen einem TCP/IP-Netzwerk auf einer Seite und einem X.25-Netzwerk auf der anderen Seite zu routen.

Auf Anfrage liefern wir Ihnen ein separates Dokument über Einrichtung der XOT-Router.

Um die XOT-Funktionalität in rvsEVO zu benutzen, brauchen Sie eine IP-Verbindung zu einem XOT-Router (z.B. CISCO 801, CISCO 2600 und C2800 oder BINTEC X4300, R1200, R4100 und R4300)

In der folgenden Tabelle werden die XOT-Parameter der lokalen Station beschrieben.

XOT-Parameter

| Parameter | Beschreibung |
|---|---|
| **eingeschaltet <enabled>** | Dieser Parameter entscheidet, ob der XOT-Empfänger gestartet werden soll, oder nicht.<br>Mögliche Werte:<br>– `Ja` (eingeschaltet)<br>– `Nein` (ausgeschaltet). |

| | |
|---|---|
| **Empfänger-Nummer <ReceiverNumber>** | **Nur bei der lokalen Station**: Nummer zur Unterscheidung der verschiedenen Empfangskanäle, über die die lokale Station erreichbar ist. Zu jeder Nummer gehört eine Registerkarte mit einem Satz von XOT-Parametern. rvsEVO vergibt und verwaltet diese Nummer automatisch.<br>**Hinweis**: Dabei entspricht die Reihenfolge der Empfängernummern der Anordnung der Registerkarten in der grafischen Oberfläche. Die XOT-Parameter in der ersten XOT-Registerkarte bekommen die Empfängernummer 1(XOT[1]); die XOT-Parameter in der zweiten XOT-Registerkarte bekommen die Empfängernummer 2 (XOT[2]) usw. Achten Sie beim Einrichten der Partnerstation darauf, über welche Empfängernummer die Kommunikation stattfinden soll. Siehe Parameter `Empfänger Nummer` (Partnerstation) in der Tabelle der Nachbarstation.<br>Standard: `1` |
| **Lokale IP-Adresse <LocalIPAddress>** | (optional): eigene IP-Adresse. Internet Adresse in der Form "255.255.255.255". Wenn Sie bei der lokalen Station keinen Wert angeben, erlauben Sie die automatische Bestimmung der Internet-Adresse. Wenn die eigene Station nur eine Internet-Adresse besitzt, sollten Sie dieses Feld freilassen. |
| **Lokaler Port <LocalPort>** | XOT-Port der lokalen Station |
| **max. eingehende Verbindungen <Sessions>** | Maximale Anzahl gleichzeitig laufender Empfangsprozesse über diesen Kanal.<br>Maximum: 100 |
| **Neustart Zeitintervall <RestartTimeout>** | Zeitintervall in Sekunden nachdem ein neuer XOT-Empfänger (Listener) starten sollte. |
| **Router IP-Adresse <RouterIPAddress>** | IP-Adresse des XOT-Routers |
| **Router Port <Port>** | Port des XOT-Routers |

| | |
|---|---|
| **Timeout**<br>**<Timeout>** | Abbruchzeit in Sekunden, nach der das Kommunikationsprogramm die Verbindung abbricht, wenn die Partnerstation nicht antwortet.<br>Standard: `30` |
| **X.25 Adresse**<br>**<X25Address>** | eigene 15-stellige X.25 Adresse.<br>Hier ist Ihre X.25-Adresse einzutragen. |
| **X.25 Dbit**<br>**<DBit>** | Das D-Bit ist ein Feld im Datenpaket von X.25, das für die Ende-zu-Ende-Bestätigung verwendet wird. Die DTE zeigt damit an, ob sie den Empfang einer Ende-zu-Ende-Bestätigung wünscht. Mögliche Werte:<br>– `true` (Ja)<br>– `false` (Nein)<br>Standard: `Nein` |
| **X.25 Facilities**<br>**<X25Facilities>** | Spezielle Informationen oder Anlagen bei X.25-Übertragung; siehe Informationen des benutzten X.25-Dienstes. |
| **X.25 Modulo**<br>**<X25Modulo>** | In der X.25-Datenübertragung wird ein auf Modulo basierendes Verfahren für die folgerichtige Übertragung von Nutzerinformationen mit Hilfe von Zählern und einem Fenstermechanismus auf der Sicherungsschicht durchgeführt. Für X.25 sind zwei Modulo-Varianten bekannt: der Modulo-8-Zähler und der Modulo-128-Zähler. Wenn Sie sich für den Modulo 128 entscheiden, bedeutet dies, dass größere Fenstergrößen gewählt werden können, die dann über den Parameter `FACILITIES` festgelegt werden. |
| **X.25 Paketgröße**<br>**<X25PacketSize>** | Größe der Datenpakete bei der Datenübertragung.<br>Standard: **128** |
| **X.25 Userdata**<br>**<X25UserData>** | Benutzerangaben für X.25-Übertragung; siehe Informationen des benutzten X.25-Dienstes |

### 3.2.3 Nachbarknoten hinzufügen und konfigurieren

Durch einen rechten Mausklick auf die lokale Station wird über Nachbarstation hinzufügen das Kontextmenü eröffnet, in dem zwischen einem TCP/IP-, TLS-, ISDN-, XOT-, Proxy TCP/IP- oder Proxy TLS-Netzwerk

gewählt werden kann. Eine TLS-Verbindung wird benötigt, wenn alle Daten, die über die Leitung gehen, verschlüsselt werden sollen.

Die möglichen Registerkarten für eine Nachbarstation sind: Nachbarstation, Key, OFTP, Leitungstyp, TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP- und Proxy TLS.

– Nachbarstation: der obligatorische Parameter in dieser Registerkarte ist SID. Der Parameter SID ist eine eindeutige PartnerID, die aus bis zu 16 Zeichen bestehen kann. Erlaubt sind: 'A-Z', '0-9', '-', '_' und '.', wobei zu beachten ist, dass '**.**' (Punkt) als erstes Zeichen nicht zugelassen ist. Der Parameter Netzwerk wird automatisch angelegt, wenn Sie eine Nachbarstation anlegen, je nachdem für welche Verbindungsart Sie sich entscheiden (TCP/IP, TLS, ISDN, XOT, Proxy TCP/IP oder Proxy TLS). Die restlichen Parameter sind optional und dienen zur Pflege der Kontaktdaten.
– Key (Zertifikatsverwaltung): In dieser Registerkarte können Sie die Zertifikate für die Dateiverschlüsselung verwalten. Nähere Informationen finden Sie im Kapitel 6 "Verschlüsselte Übertragung mit rvsEVO (Dateiverschlüsselung)"
– OFTP: Dieser Registerkarte beinhaltet Parameter, die sich auf das ODETTE-Protokoll beziehen. In der folgenden Tabelle werden Sie beschrieben.

OFTP-Parameter

| Parameter | Beschreibung |
|---|---|
| **Austauschpuffergröße <BufferSize>** | Maximale Größe des Übertragungspuffers (Exchange Buffer Size) in Bytes.<br>Mögliche Werte: 0 - 99999 (Default:10.000).<br>**Achtung:** Wenn Sie den Datenaustausch über ein ISDN-Netz betreiben, sollte der Wert nicht über 4000 Bytes liegen, da sonst die Übertragung großer Dateien Probleme verursachen kann. |
| **Austauschpuffer-Credit <Credit>** | Maximale Zahl der gesendeten Blöcke (Exchange Buffer) ohne Erwartung einer Quittung.<br>Mögliche Werte: 0 - 999 (Default: 999). |
| **Authentifizierung <Authentication>** | Gegenseitige Authentifizierung auf ODETTE-Ebene, die am Anfang einer Übertragung stattfinden soll. Mögliche Werte:<br>- `Ja` (X.509-Zertifikate müssen ausgetauscht sein)<br>- `Nein` |

| | |
|---|---|
| **EERP ausgehend<br><EERPOut>** | Handhabung für das Senden von Empfangsbestätigungen (EERPs/NERPs).<br>NORMAL: Generieren einer Empfangsbestätigung nach erfolgreichem Empfang einer Datei und unmittelbares, aktives Versenden.<br>NEVER: es wird keine Empfangsbestätigung versendet. Der Status des Jobs wird auf ENDED gesetzt.<br>HOLD eine Empfangsbestätigung wird generiert, jedoch nicht versandt. Um versandt zu werden, muss die Empfangsbestätigung zuerst manuell mit dem Programm handleEERP (oder in der GUI mit der Schaltfläche Freigeben; Maske Sendeübertragung) freigegeben werden.<br>SYNC: die OFTP-Session wird so lange aufrecht erhalten, bis die Empfangsbestätigung vom Empfänger beim Sender angekommen ist.<br>ROUTING_SYNC: **Hinweis**: Dieser Parameter ist nur für ein rvsEVO, welches als Routing-Station dient, von Bedeutung. Dabei wird die OFTP-Session zwischen dem Router und dem Sender so lange aufrecht erhalten, bis die Empfangsbestätigung vom Empfänger angekommen ist, um diese dann in der selben OFTP-Session an den Sender zurückzusenden.<br>Standard: NORMAL<br>(S. auch Parameter **EERP eingehend**) |
| **EERP eingehend<br><EERPIn>** | Handhabung für das Empfangen von Empfangsbestätigungen (EERPs/NERPs).<br>NORMAL: Sendeübertragung wird nach Erhalt eines EERPs beendet.<br>NEVER: der Status des Sendejobs wird auch ohne EERP-Eingang auf ENDED gesetzt.<br>Standard: NORMAL<br>**Hinweis**: Es besteht die Möglichkeit, dass die Angabe von EERP Destination und EERP Originator bei Ihrem Partner vertauscht ist. Dies kann zu Problemen bei der Erkennung von EERPs/ NERPs führen. Mit der Einstellung des Parameters rvs_evo.job.additional_erp_reverse_search=true in der Datei $RVS_HOME\conf\rvs-system.properties können Sie dieses Problem umgehen. Bei Nichterkennung eines EERPs/NERPs wird auch in der entgegengesetzten Richtung gesucht. |
| **Empfangspasswort<br><PasswordReceive>** | Das Passwort, das rvsEVO von der Nachbarstation erwartet. |

| | |
|---|---|
| **ERP-Signatur**<br>**<SignERP>** | Für die Station festgelegter Sendeparameter, der beim Erstellen eines Sendeauftrags nicht überschrieben werden kann. Steht nur zur Verfügung, wenn Parameter **File Service Proxy = Extern**. Die Beschreibung entnehmen Sie bitte dem Parameter **signierten EERP/NERP beantragen** der Tabelle "Sendeparameter” auf Seite 103. |
| **File Service Proxy**<br>**<Proxy>** | Nur für File Service Modul; mögliche Werte:<br>– kein (Standard)<br>– Intern<br>– Extern<br>Für weitere Informationen sehen Sie bitte Kapitel 8 "File Service Modul" |
| **Komprimierung**<br>**<Compression>** | Für die Station festgelegter Sendeparameter, der beim Erstellen eines Sendeauftrags nicht überschrieben werden kann. Steht nur zur Verfügung, wenn Parameter **File Service Proxy = Extern**. Die Beschreibung entnehmen Sie bitte dem Parameter **Offline Komprimierung** der Tabelle "Sendeparameter” auf Seite 103. |
| **OFTP-Version**<br>**<level>** | Hier ist die anzuwendende ODETTE-Protokollversion festzulegen. Normalerweise muss ein OFTP-Produkt in der Lage sein, während einer Odette-Session die Odette-Protokollversion auszuhandeln. Wenn dies Ihr Partner nicht kann, sollen Sie die ODETTE-Protokollversion über diesen Parameter fest einstellen. Mögliche Werte : 1.2,1.3, 1.4, 2. Standard: **2** |
| **ODETTE-ID**<br>**<OdetteID>** | Im Gegensatz zur Stations-ID, die nur auf Ihrem Rechner eindeutig sein muss, ist die **ODETTE** ID eine weltweit eindeutige Identifikation der Partner- oder eigenen Station, wenn das ODETTE File Transfer Protokoll benutzt wird.<br>Die **ODETTE** ID besteht aus 25 Zeichen, bestehend aus dem Buchstaben "O", der 18-stelligen Organisations-Identifikation, die Sie beim VDA (Verband der Automobilindustie) beantragen können und der 6-stelligen Computer-Adresse innerhalb der jeweiligen Organisation. Die Computer-Adresse ist von der Organisation frei wählbar und muss eindeutig sein. |
| **PKI** | für genaue Informationen sehen Sie bitte Kapitel 11.2.1 „Konfiguration der Stationsliste“ |
| **Sendepasswort**<br>**<PasswordSend>** | Das Passwort, das rvsEVO der Nachbarstation sendet. Die ODETTE-Passwörter werden immer zwischen den Nachbarstationen während des Aufbaus einer Session ausgetauscht und überprüft. |

| | |
|---|---|
| **SFIDDESC als Dateiname <UseDescAsFilename>** | SFID ( Start File ID)-Description wird als Dateiname (anstatt VDSN) für die Übertragung verwendet; das Feld SFIDDESC wird in rvsEVO beim Erzeugen eines Sendeauftrags mit dem Parameter `File Description` gefüllt. Mögliche Werte: true (Ja) oder false (Nein); Standard: false |
| **Sicherheit <SecuritySet>** | Dieser Parameter regelt, wie die Dateiverschlüsselung beim Datentrasfer angewandet wird.<br>- `NEIN` (`NO`): Verschlüsselung nicht möglich. Falls ein Sendejob Verschlüsseung verlangt, wird der Job mit einer Fehlermeldung abgebrochen.<br>- `OPTIONAL` (`OPT`): Verschlüsselung optional möglich und kann im Sendejob festgelegt werden.<br>- `ZWINGEND` (`FORCED`): Verschlüsselung zwingend. Falls ein Sendejob die Verschlüsselung nicht vorsieht, wird eine Fehlermeldung ausgegeben. Wenn eine Partnerstation eine Datei unverschlüsselt sendet, wird der Empfang der Datei abgelehnt.<br>Standard: `SICHERHEIT` = `OPTIONAL`. |
| **Sicherheitsmerkmale <SecurityFeatureSet>** | Für die Station festgelegter Sendeparameter, der beim Erstellen eines Sendeauftrags nicht überschrieben werden kann. Steht nur zur Verfügung, wenn Parameter **File Service Proxy = Extern**. Die Beschreibung entnehmen Sie bitte der Tabelle "Sendeparameter” auf Seite 103. |
| **Signatur <Sign>** | Für die Station festgelegter Sendeparameter, der beim Erstellen eines Sendeauftrags nicht überschrieben werden kann. Steht nur zur Verfügung, wenn Parameter **File Service Proxy = Extern**. Die Beschreibung entnehmen Sie bitte dem Parameter **Dateisignatur** der Tabelle "Sendeparameter” auf Seite 103. |
| **VDSN-Zeichensatz <VdsnCharset>** | Betrifft nur Dateiempfang: Zeichensatz der für den virtuellen Dateinamen zugelassen wird. Mögliche Werte:<br>- `ODETTE`: nur der ODETTE-Zeichensatz ist zulässig (Standard)<br>- `ALL`: alle ASCII-Zeichen sind zulässig |
| **Verschlüsselung <Encryption>** | Für die Station festgelegter Sendeparameter, der beim Erstellen eines Sendeauftrags nicht überschrieben werden kann. Steht nur zur Verfügung, wenn Parameter **File Service Proxy = Extern**. Die Beschreibung entnehmen Sie bitte der Tabelle "Sendeparameter” auf Seite 103. |

| | |
|---|---|
| **Verschlüsselungsalgorithmus <EncryptionAlgorythm>** | Für die Station festgelegter Sendeparameter, der beim Erstellen eines Sendeauftrags nicht überschrieben werden kann. Steht nur zur Verfügung, wenn Parameter **File Service Proxy = Extern**. Die Beschreibung entnehmen Sie bitte der Tabelle "Sendeparameter” auf Seite 103. |
| **Wiederaufsatz <Restart>** | Mit diesem Parameter legen Sie fest, ob ein fehlgeschlagener Job wieder angestoßen werden darf.<br>Mögliche Werte:<br>true (Ja / standard) oder false (Nein) |
| **Zertifikatsvalidierung <Certificatevalidati-onType>** | Für genaue Informationen sehen Sie bitte Kapitel 11.2.1 „Konfiguration der Stationsliste“ |

– Leitungstyp: In der Registerkarte **Leitungstyp** befindet sich nur der Parameter `Aktiver Verbindungsaufbau`. Mit diesem Parameter können Sie festelegen, ob Ihre lokale Station zu der in diesem Parametersatz vereinbarten Station aktiv die Verbindung aufbauen soll oder nicht. `Ja` bedeutet, dass sobald die Sendeaufträge erzeugt werden, die Verbindung zur jeweiligen Station automatisch aufgebaut wird und Dateien versendet werden; bei `Nein` werden anstehende Sendeauträge nicht automatisch versendet, sondern es wird gewartet, bis die Gegenstelle die Verbindung aktiviert. Erst wenn die Verbindung steht, werden Dateien versendet.

– TCP/IP und TLS: Die TCP/IP und die TLS-Registerkarte haben die gleichen Parameter.

TCP/IP-Parameter
TLS-Paramter

| Parameter | Beschreibung |
|---|---|
| **IP-Adresse** | IP-Adresse oder DNS-Name der eigenen Station für die TCP/IP- oder TLS-Verbindung. |
| **Nummer des zugehörigen Empfängers** | Nicht aktiv! Nummer des Empfangskanals, wenn für die lokale Station mehrere Empfängers konfiguriert wurden (mehrere Registerkarten). Über diese Nummer ist es möglich festzulegen, über welche Empfänger-Nummer die Kommunikation für die betreffende Station stattfindet. Die TCP/IP- und TLS-Empfänger werden separat hochgezählt. Wenn für die lokale Station nur ein Empfänger definiert wurde, ist dies die `Empfänger Nummer 1` (Standard) |

| | |
|---|---|
| **Port** | Port auf dem ein Listener gestartet werden soll; standardmäßig 6619. |
| **Proxy Bastion** | nur für rvs® OFTP Proxy; aus einer Liste, der in rvsEVO angelegten Bastion Instances, können Sie Bastion auswählen, über die Sie kommunizieren möchten. |

In der nächsten Tabelle werden die ISDN-Parameter für eine Partnerstation erklärt:

ISDN-Parameter

| Parameter | Beschreibung |
|---|---|
| **Anzahl Wahlwiederholungen** | Anzahl der Wahlwiederholungen, wenn der Partner nicht antwortet. |
| **ISDN Facilities <Facilities>** | Spezielle Informationen oder Anlagen der ISDN-Übertragung. |
| **ISDN Nummer <ISDNNumber>** | ISDN-Nummer unter der die Partnerstation erreichbar ist. |
| **ISDN Protokoll <Protokol>** | Gibt den genutzten ISDN-Standard an.<br>- 1TR6: Deutscher Landesstandard<br>- E-DSS1: EURO-ISDN<br>Standard: **E-DSS1** |
| **ISDN Terminal Identifier** | Nur nötig für X.31: **T**erminal **E**nd **I**dentification.<br>Standard: **0** – keine TEI vergeben |
| **ISDN Userdata** | Benutzerangaben für die ISDN-Übertragung |
| **Kartennummer** | Nummer der im Rechner installierten ISDN-Karte, beginnend mit "0" |
| **Nummer des zugeordneten Empfängers <ReceiverNumber>** | Nummer des Empfangskanals, wenn für die lokale Station mehrere Empfängers konfiguriert wurden (mehrere Registerkarten). Über diese Nummer ist es möglich festzulegen, über welche Empfänger-Nummer die Kommunikation für die betreffende Station stattfindet. Wenn für die lokale Station nur ein Empfänger definiert wurde, ist dies die Empfänger Nummer 1 (Standard) |
| **Typ** | CAPI2A |

| | |
|---|---|
| **Verbindungs-Timeout** | Abbruchzeit in Sekunden, nach der das Kommunikationsprogramm die Verbindung abbricht, wenn die Partnerstation antwortet.<br>Standard: **30** |
| **Wartezeit Wahlwiederholungen** | Wartezeit zwischen zwei Wahlwiederholungen, wenn der Partner nicht antwortet. |
| **X.25 Adresse** | 15-stellige X.25 DTE Adresse der Partnerstation. Angabe ist optional. Es wird jedoch empfohlen, hier die ISDN Nummer einzutragen, da manche Gegenstellen eine X.25 Adresse erwarten. |
| **X.25 Dbit** | Das D-Bit ist ein Feld im Datenpaket von X.25, das für die Ende-zu-Ende-Bestätigung benutzt wird. Die DTE zeigt damit an, ob sie den Empfang einer Ende-zu-Ende-Bestätigung wünscht.<br>Standard: `Nein` |
| **X.25 Facilities** | Spezielle Informationen oder Anlagen bei X.25-Übertragung; siehe Informationen des benutzten X.25-Dienstes. |
| **X.25 Fenstergröße** | Die Fenstergröße bei X.25/ISDN-Kommunikation ist die Anzahl der Pakete, die ohne Bestätigung verschickt werden können. Die Fenstergröße kann während des Verbindungsaufbaus ausgehandelt werden. Wir empfehlen aber von vornherein die zum Partnernetzwerk passende Fenstergröße zu konfigurieren. `7` ist empfholen für ISDN und `2` für X.25 native. |
| **X.25 geschlossene Benutzergruppe (closed user group)** | ISDN und X.25 bieten die Möglichkeit, eine geschlossene Benutzergruppe zu bilden. Alle Teilnehmer, die zu einer solchen Gruppe gehören, können untereinander über das öffentliche Telekommunikationsnetz kommunizieren. Verbindungsanforderungen an Gruppenmitglieder von Teilnehmern, die nicht Mitglied der geschlossenen Benutzergruppe sind, werden von der Vermittlungsstelle abgewiesen. Dies gilt gleichermaßen für Verbindungsanforderungen von Gruppenmitgliedern an Teilnehmer, die nicht Mitglied der geschlossenen Benutzergruppe sind. Dieses Dienstemerkmal wird als Closed User Group (CUG) bezeichnet.<br>Standard: **Nein** |
| **X.25 Paketgröße** | Größe der Datenpakete bei der Datenübertragung.<br>Standard: **128** |
| **X.25 Userdata** | Benutzerangaben für X.25-Übertragung; siehe Informationen des benutzten X.25-Dienstes |

### XOT

**Hinweis**: Auf Anfrage können wir Ihnen ein Dokument mit der Beschreibung der XOT-Routerkonfiguration zusenden.

In der nächsten Tabelle werden die XOT-Parameter für eine Partnerstation erklärt:

XOT-Paramter

| Parameter | Beschreibung |
|---|---|
| **Nummer des zugeordneten Empfängers** | Nummer des Empfangskanals, wenn für die lokale Station mehrere Empfängers konfiguriert wurden (mehrere Registerkarten). Über diese Nummer ist es möglich festzulegen, über welche Empfänger-Nummer die Kommunikation für die betreffende Station stattfindet. Wenn für die lokale Station nur ein Empfänger definiert wurde, ist dies die `Empfänger Nummer 1` (Standard) |
| **Router IP-Adresse** | IP-Adresse des XOT-Routers |
| **Router Port** | Port des XOT-Routers |
| **Timeout** | Abbruchzeit in Sekunden, nach der das Kommunikationsprogramm die Verbindung abbricht, wenn die Partnerstation nicht antwortet. Standard: **30** |
| **X.25 Adresse** | 15-stellige X.25 DTE Adresse der Partnerstation. |
| **X.25 Dbit** | Das D-Bit ist ein Feld im Datenpaket von X.25, das für die Ende-zu-Ende-Bestätigung benutzt wird. Die DTE zeigt damit an, ob sie den Empfang einer Ende-zu-Ende-Bestätigung wünscht. Standard: `Nein` |
| **X.25 Facilities** | Spezielle Informationen oder Anlagen bei X.25-Übertragung; siehe Informationen des benutzten X.25-Dienstes. |

| | |
|---|---|
| **X.25 Modulo** | In der X.25-Datenübertragung wird ein auf Modulo basierendes Verfahren für die folgerichtige Übertragung von Nutzerinformationen mit Hilfe von Zählern und einem Fenstermechanismus auf der Sicherungsschicht durchgeführt. Für X.25 sind zwei Modulo-Varianten bekannt: der Modulo-8-Zähler und der Modulo-128-Zähler. Wenn Sie sich für den Modulo 128 entscheiden, bedeutet dies, dass größere Fenstergrößen gewählt werden können, die dann über den Parameter `FACILITIES` festgelegt werden. |
| **X.25 Paketgröße** | Größe der Datenpakete bei der Datenübertragung.<br>Standard: **128** |
| **X.25 Userdata** | Benutzerangaben für die ISDN-Übertragung |

Ein rechter Mausklick auf einen Nachbarknoten (eine Nachbarstation) öffnet das Kontextmenü mit den Optionen: **geroutete Station hinzufügen, Eintrag entfernen** und **Verbindung aktivieren**. Mit der Option **Verbindung aktivieren** können Sie die Nachbarstation aktivieren, um die Dateien, die für Ihre Station bestimmt sind, abzuholen. Mit der Option **geroutete Station hinzufügen** können Sie eine geroutete Station anlegen, die über die direkte Nachbarstation erreichbar ist. Mit dem **Eintrag entfernen** können Sie eine schon angelegte geroutete Station löschen.

**Hinweis**: Ihre lokale Station kann über verschiedene Datenleitungen und Leitungsprotokolle von entfernten Stationen Daten empfangen. Dies bedeutet, dass es möglich ist, mehrere Empfänger vom gleichen Netzwerktyp oder Empfänger von verschiedenen Netzwerktypen für die lokale Station zu konfigurieren.

### 3.2.4 Geroutete Stationen hinzufügen und konfigurieren

Voraussetzung

Sie haben einen direkten Nachbarknoten, über den Sie diese geroutete Station erreichen können, schon eingerichtet.

Klicken Sie mit der rechten Maustaste auf die Nachbarstation und wählen im Kontextmenü den Eintrag **geroutete Station hinzufügen**. Ein neuer Pfeil zeigt auf die neue Station. Für Sie ist es nicht von Bedeutung, mit welcher Verbindungsart diese Station erreicht wird (das regelt die Nachbarstation). Deswegen ist auf diesem Pfeil auch keine Verbindungsart zu sehen.

Die möglichen Registerkarten für eine geroutete Station sind: Geroutete Station und OFTP.

Die obligatorischen Parameter in der Registerkarte Geroutete Station sind SID und Nachbarstation.

– SID: StationID der gerouteten Station

- Nachbarstation: Eine geroutete Station kann nur über eine direkte Nachbarstation erreicht werden, da es zu einer gerouteten Station keine direkte Netzwerkverbindung gibt. Die Information über welche Nachbarstation diese geroutete Station zu erreichen ist, ist in diesem Parameter hinterlegt.

Dieser Parameter entspricht dem Parameter `Gateway` im Element `StationRouted` in der XML-Stationskonfigurationsdatei.

In der Registerkarte OFTP sind die ODETTE-Parameter. Der obligatorische Parameter in dieser Registerkarte ist nur `Odette Id`. Lesen Sie bitte den Abschnitt **Nachbarknoten konfigurieren** in diesem Kapitel für die Erklärung der Parameter `Odette Id`, `EERP ausgehend` und `OFTP-Version`.

Löschen von gerouteten Stationen

Eine Station löschen können Sie mit der Option **Eintrag entfernen.** Zu dieser Option gelangen Sie, indem Sie mit einem Klick der rechten Maustaste das Kontextmenü einer markierten Station öffnen.

### 3.2.5 Virtuelle Stationen hinzufügen und konfigurieren

Was sind virtuelle Stationen?

Virtuelle Stationen dienen als Repräsentation von Stationen außerhalb des OFTP-Netzwerkes für das OFTP-Netzwerk. Sie ermöglichen das Versenden von Dateien an Zielstationen, die außerhalb des OFTP-Netzwerkes liegen.

Innerhalb eines OFTP-Netzwerkes wird die eindeutige Identifikation einer Station über die ODETTE-ID festgelegt. Auch jeder virtuellen Station muss eine eigene ODETTE-ID zugeordnet werden. Eine virtuelle Station kann senden und empfangen.

Virtuelle Station hinzufügen

Durch einen rechten Mausklick auf die lokale Station können Sie über **Virtuelle Stationen hinzufügen** eine virtuelle Station anlegen.

Die Registerkarten für eine virtuelle Station sind:

- Virtuelle Station: der obligatorische Parameter in dieser Registerkarte ist SID. Der Parameter SID ist eine eindeutige PartnerID, die aus bis zu 16 Zeichen bestehen kann. Die restlichen Parameter sind optional und dienen der Pflege der Kontaktdaten.
- OFTP (ODETTE-Parameter): der obligatorische Parameter in dieser Registerkarte ist die Odette-ID. Weitere Informationen über die Odette-ID finden Sie im Kapitel 3.2.2 "Lokale Station konfigurieren". Die Beschreibung der Parameter PKI und Zertifikatsvalidierung finden Sie im Abschnitt 11 "PKI-Anbindung mit rvsEVO".

Löschen von virtuellen Stationen

Sie können eine Station löschen, indem Sie die Station markieren, mit der rechten Maustaste das Kontextmenü offnen und anschließend **Eintrag entfernen** wählen.

**Hinweis:** Bei der Partnerstation wird die virtuelle Station wie eine geroutete Station behandelt.

### 3.2.6 Konfiguration der Verschlüsselungskomponente bei einem TLS-Empfänger

In diesem Kapitel wird beschrieben, welche Schritte notwendig sind zur Einrichtung verschlüsselter Kommunikation mittels TLS (Leitungsverschlüsselung). Besonders eingegangen wird auf die Parameter **Schlüsselverwaltungsdatei** und **Schlüsseldateiname für vertrauenswürdige Zertifikate**, ebenso wie auf die Funktionen, die sich über diese Parameter steuern lassen. Eine Beschreibung der restlichen TLS-Parameter finden Sie im Kapitel 3.2 "Anpassen der Stationskonfiguration".

**Hinweis**: Über die Remote-GUI kann die Verschlüsselungskomponente eines TLS-Empfängers nicht konfiguriert werden.

Ein rechter Mausklick auf die lokale Station öffnet das Kontextmenu mit der Möglichkeit, einen TLS-Empfänger hinzuzufügen. Bitte füllen Sie die ersten Parameter, wie im Kapitel 3.2 beschrieben, aus.

#### Welche Schritte sind für eine verschlüsselte Kommunikation notwendig?

Die nächsten Schritte sind die Voraussetzung für verschlüsselte Kommunikation mittels TLS.

- Generieren Sie Ihr eigenes Schlüsselpaar.
- Exportieren Sie den eigenen öffentlichen Schlüssel in Form von X.509 Zertifikaten.
- Senden Sie Ihren eigenen öffentlichen Schlüssel zu Ihrem Partner.
- Ihr Partner muss Ihren öffentlichen Schlüssel (public key) in sein OFTP-Produkt importieren und Ihnen sein X.509-Zertifikat zusenden.
- Wenn sie das X.509-Zertifikat vom Partner erhalten haben, importieren Sie es in Ihre Schlüsseldatei für vertrauenswürdige Zertifikate (`TlsTrustManager`).

#### Wie generiere ich ein eigenes Schlüsselpaar?

Nachdem Sie einen TLS-Empfänger Ihrer lokalen Station hinzufügt haben, müssen Sie zuerst Ihr eigenes Schlüsselpaar für die TLS-Kommunikation generieren.

- Klicken Sie auf den Wert des Parameters **Schlüsseldateiname** in der Maske des TLS-Empfängers.
- Diese Aktion macht zwei Symbole sichtbar: **...** und **-->**.
- Das Symbol **...** öffnet den Dialog, in dem Sie eine andere Schlüsselverwaltungsdatei und nicht die Standarddatei `$RVS_HOME/system/data/tlsKeyManagerKeyStore.p12` wählen können. Wir empfehlen, die Standarddatei zu verwenden.

- Das Symbol **-->** startet das Programm Portecle. Dieses Programm soll für die Verwaltung von Schlüsseln verwendet werden.

So generieren Sie ein Schlüsselpaar:

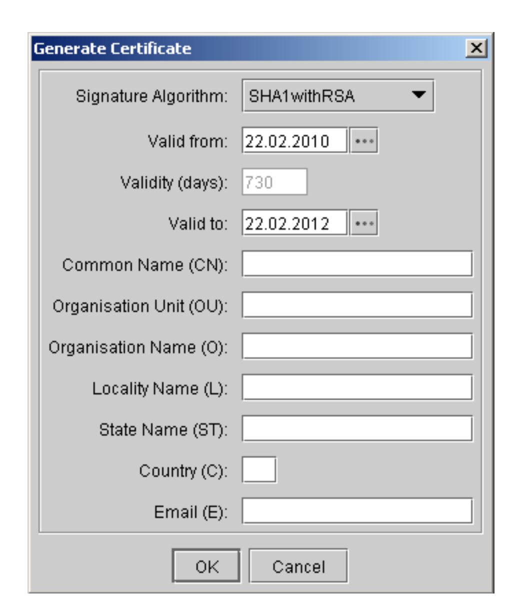
– Starten Sie Portecle mit **-->**

– Um zur Funktionalität **Schlüsselpaar erzeugen** (**Generate Key Pair**) zu gelangen, wählen Sie das Symbol **Tools** in der Funktionsleiste.

– Im Fenster **Generate Key Pair** wählen Sie als Schlüssel-Algorithmus `RSA` und als Standard-Schlüssellänge (default key size) `1024`.

– Drücken Sie die **OK**-Schaltfläche und das Fenster **Generate Certificate** öffnet sich.

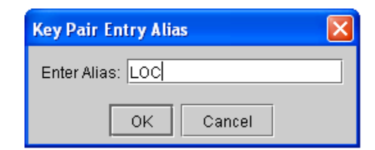
– In diesem Fenster sind folgende Felder zu setzen:




Der Parameter `Common Name` ist z.B. von Bedeutung, wenn es sich um eine Anbindung an eine vorhandene PKI (Public Key Infrastructure) handelt und diese Angabe für die LDAP-Struktur benötigt wird. Dieser Parameter ist obligatorisch.

Die restlichen Angaben betreffen Ihre Organisation. rvsEVO macht hier keine Vorschriften, wie diese Felder auszufüllen sind.

– Nach dem Ausfüllen aller Felder und Bestätigung mit **OK** öffnet sich das nächste Fenster.



Das Feld **Key Pair Entry Alias** ist nur für interne Zwecke, um Zertifikate zu kennzeichnen. Wichtig ist, dieses Feld nicht leer zu lassen. Sie können das Feld mit Common Name, wie vorgeschlagen oder mit Ihrer StationsID füllen.

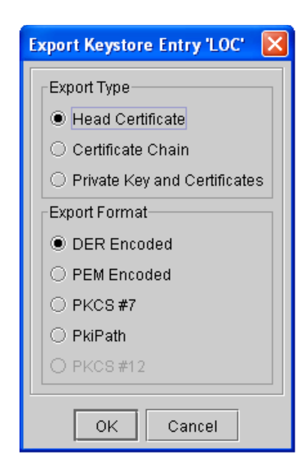
– Nach Drücken von **OK** soll ein Schlüsselpaar erfolgreich erzeugt werden.

**Export des eigenen öffentlichen Schlüssels als Zertifikat**

– Rufen Sie wieder Portecle auf, indem Sie zuerst auf den Wert des Parameters **Schlüsselverwaltungsdatei** klicken und dann auf das Symbol **-->**.

– Klicken Sie auf Ihr Schlüsselpaar mit der rechten Maustaste. Das Kontextmenü öffnet sich und bietet Ihnen die Funktion **Export**.

– Mit Aktivierung dieser Option des Kontextmenüs öffnet sich das nächste Fenster **Export Keystore Entry**.




– Verwenden Sie die Standardeinstellungen für Export: **Export Type**: `Head Certificate`; **Export Format**: `DER Encoded`.

– Bestätigen Sie Ihre Angaben mit **OK**.

– Im nächsten Fenster sollten Sie den Namen, unter welchem das Zertifikat gespeichert wird, eingeben. Wir empfehlen, dass dieser Name mit `.cer` oder `.crt` endet. **Hinweis**: Erstellen Sie auch eine Kopie Ihrer Daten für die TLS-Verschlüsselung und speichern Sie diese an einem sicheren Ort.

**Import eines X.509-Zertifikats eines Partners**

– Rufen Sie wieder Portecle auf, indem Sie zuerst auf den Wert des Parameters **Schlüsseldateiname für vertrauenswürdige Zertifikate** und dann auf das Symbol **-->** klicken .

– Wählen Sie das **Tool** icon in der Funktionsleiste and dann die Funktion **Import Trusted Certificate**.

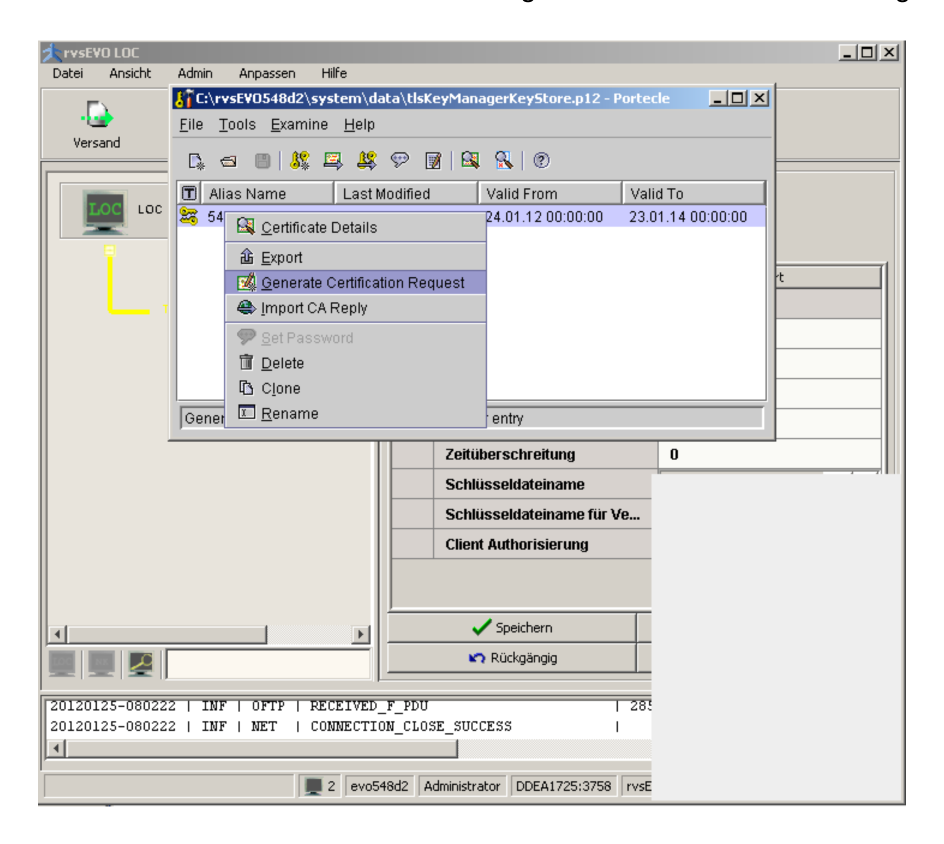
– Wählen Sie die Zertifikatsdatei im nächsten Fenster.

### Vorgehensweise bei Verwendung von CA-Zertifikaten

CA-Zertifikate (**C**ertificate **A**uthority) werden von einem zertifizierten Dienstleister (TrustCenter), z. B. der Odette Zertifizierungsstelle, ausgestellt. Bitte gehen Sie wie folgt vor, wenn Sie mit CA-Zertifikaten arbeiten:

CSR generieren

- Generieren Sie ein Schlüsselpaar, wie in diesem Kapitel beschrieben.
- Generieren Sie aus Ihrem öffentlichen Schlüssel eine Zertifikatsregistrierungsanforderung (CSR - **C**ertificate **S**igning **R**equest), indem Sie mit der rechten Maustaste auf Ihr Schlüsselpaar klicken (über Schlüsseldateiname und --> erreichbar) und anschließend die Option `Generate Certification Request` wählen. Name und Verzeichnis, in dem der Request abgelegt werden soll, sind frei wählbar. Die Datei wird standardmäßig im Format PKCS#10 erzeugt.




Beispiel eines Zertifizierungsrequests:


```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBWDCBwgIBADAbMRkwFwYDVQQDDBB1dm81NDhkLW9yaS1maWx1MIGfMA0GCSqGS
A4GNADCBiQKBgQCczTte+HyTxy/1CCjtXWuADqO7H/LDDKfH7GZANKS2WqJ9+Ozw7
Tcik5JS3Re8L3q6xGBzDxucD7N6Qp3ikincYTxaE92yuy+rbK43YJJRLdgixKAu+a
YY/a2UjKOeGY+oVRmWJL1u+pbIs9VAStMwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBA
p426CHVOREM6QzVWf2qNe4OR/mxhuKT118kP1xx9oG4K7pkPbHiMuT1SdNmzb/GDp
tisBm185o+fCDOKaOpIKidjC9aTact16YtSoRdc4v7RIw5t1zpRw+ANhwOwVwZMRg
/pK79MbM
-----END NEW CERTIFICATE REQUEST-----
```

- Senden Sie den Zertifikatsrequest an Ihren Zertifizierungsdienstleister. Bei der Odette-Zertifizierungsstelle können Sie den Antrag Online stellen: `https://www.odetteca.com`. Ausführliche Informationen finden Sie im Internet: `https://forum.odette.org/repository/odette-ca-help.pdf`.

Der Antrag wird von der Zertifizierungsstelle überprüft, welche bei einem positiven Ergebnis ein signiertes Serverzertifikat erstellt. Anschließend erhalten Sie von der CA neben Ihrem Serverzertifikat zwei weitere Zertifikate: ein CA-Root- und ein CA-Zertifikat.

eigene Zertifikate importieren

Importieren Sie zuerst das CA-Root- und anschließend das CA-Zertifikat in die Datei `$RVS_HOME/system/data/tlsKeyManagerKeyStore.p12` .

- Starten Sie Portecle mit **-->**.
- Die Datei `$RVS_HOME/system/data/tlsKeyManagerKeyStore.p12` wird standardmäßig geöffnet.
- Importieren Sie zuerst das CA-Root-Zertifikat mit der Funktion **Tools -> Import Trusted Certificate**
- Importieren Sie danach das CA- Zertifikat mit der Funktion **Tools -> Import Trusted Certificate**
- Importieren Sie anschließend das Serverzertifikat mit der Funktion **Import CA Reply** (importiere CA Antwort), indem Sie das Schlüsselpaar markieren, für welches die CA-Reply importiert werden soll, und nach Betätigung der rechten Maustaste **Import CA Reply** auswählen.
- Speichern Sie den Keystore und beenden Sie diesen anschließend.

**Wichtig:** Bitte halten Sie beim Import der Zertifikate unbedingt die angegebene Reihenfolge ein!

Senden Sie diese Zertifikate auch Ihrem Partner zu und importieren Sie ebenfalls die Zertifikate Ihres Partners wie folgt:

Zertifikate des Partners importieren

- Öffnen Sie über Portecle die Datei für die vertrauenswürdigen Zertifikate `$RVS_HOME/system/data/tlsTrustManagerKeyStore.p12` (über **Schlüsseldateiname für vertrauenswürdige Zertifikate** und **->**).
- Importieren Sie zuerst das CA-Root-Zertifikat mit der Funktion **Tools -> Import Trusted Certificate**
- Importieren Sie danach das CA-Zertifikat mit der Funktion **Tools -> Import Trusted Certificate**

**Hinweis:** Für TLS-/Proxy-TLS ist es ausreichend, wenn Sie nur das CA-Root und das CA-Zertifikat vom Partner importieren; Serverzertifikate werden automatisch während des Aufbaus der TLS-/Proxy-TLS-Verbindung beim Handshake (auf der Netzwerkebene) ausgetauscht.

Um zu erfahren, wie Sie nach der Konfiguration eine Datei versenden können, lesen Sie bitte das Kapitel 4.5 "Versand einer Datei"

### 3.2.7 Konfiguration von Stationen mittels XML-Datei

Alternativ zur grafischen Oberfläche kann die Stationskonfiguration auch mittels XML-Datei durchgeführt werden.

rvsStation list.xml

Diese Änderungen können direkt in der Datei rvsStationlist.xml vorgenommen werden. Das Element StationLoc aus der Stationskonfigurationsdatei ist äquivalent zu der Lokalen Station in der GUI, StationNeighbour ist die Nachbarstation, StationRouted die Geroutete Station und StationVirtual ist die virtuelle Station.

Auszug aus der Datei rvsStationlist.xml:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RVS_STATION_CONFIG>
-<StationLoc>
 . . .
-</StationLoc>
-<StationNeighbour>
 . . .
-<StationNeighbour>
-<StationRouted>
. . .
-</StationRouted>
-<StationVirtual>
. . .
-</StationVirtual>
</RVS_STATION_CONFIG>
```

Kommentare

Bereiche, die mit dieser Zeichenfolge (<!--) beginnen und dieser Zeichenfolge (-->) enden, werden als Kommentar interpretiert.

Benutzen Sie zum Editieren der Datei ein Textbearbeitungsprogramm (z.B. Edit, TextPad). Achten Sie bitte bei der Bearbeitung der XML-Dateien darauf, dass Sie diese als gültiges XML-Dokument ablegen, sonst kann rvsEVO diese nicht lesen und eventuell nicht richtig starten.

**Hinweis**: Sie müssen dafür sorgen, dass die IP-Ports: RMI-Port (1099) und der Odette-Port (z.B. 3305) frei sind, damit die TCP/IP-Kommunikation einwandfrei funktioniert.

Die Stationsparameter sind in Kapitel 3.2.1 beschrieben. Der Name des XML-Elements, das den Parameter repräsentiert wurde immer in spitzen Klammern aufgeführt.

updateStation-List

Zusätzlich besteht die Möglichkeit die Stationen zur Laufzeit mit dem Kommandozeilentool updateStationList zu verwalten. Hierzu muss eine XML-Datei erzeugt werden, die alle Informationen über die zu ändernden Stationen enthält. Der Inhalt dieser Datei ist in einem XML-Schema beschrieben ($RVS_HOME/system/data/xsd/stationList.xsd). Für jede Station, die in der XML-Datei aufgeführt ist, ist ein zusätzliches XML-

Element namens `Funktion` erforderlich, in dem die auszuführende Aktivität gesteuert wird.

Mögliche Direktiven:

– `insert`: Hinzufügen neuer Partnerstationen zur Stationsliste (es darf keine SID doppelt angelegt werden)
– `update:` Aktualisieren vorhandener Stationen
– `delete`: Löschen einer Station (die eigene Station kann nicht gelöscht werden)

Diese Fuktionalität wird mit folgendem Tool ausgeführt:

```
updateStationList -f <xml-filename>
```

In der GUI können können Sie mit dem Menüpunkt `Datei/Aktualisieren Stationslistenansicht` die Stationliste aktualisieren. Ein Neustart ist nicht mehr erforderlich.

## 3.3 Konfiguration der Jobstarts

Jobstarts

Die Jobstarts beinhalten Regeln, die dazu führen, dass bei entsprechenden ein- bzw. ausgehenden Dateien spezielle Programme gestartet werden können. Sie entsprechen den Residenten Empfangseinträgen (Jobstarts nach Empfang) und den Jobstarts nach Sendeversuchen in rvs®. Ab rvsEVO Version 5.03 können Sie zusätzlich Jobstarts konfigurieren, die nach einem Jobverarbeitungsfehler gestartet werden.

**Hinweis**: Sichern Sie eigene oder geänderte Skripte nicht im Verzeichnis `$RVS_HOME\bin` oder dessen Unterverzeichnisse, da diese während einer Update-Installation überschrieben werden.

Jobfilter

Falls mehr als ein Jobfilter auf einen Job zutrifft, so wird das Programm des am meisten zutreffenden Jobfilters gestartet. Bei der Verwendung von Platzhaltern ist z.B. Testdatei* genauer als Test*.

Die Jobstarts können über die rvsEVO-GUI oder in der XML-Jobstarts-Konfigurationsdatei `rvsJobstart.xml` konfiguriert werden.

**Hinweis**: Nach der Installation von rvsEVO sind zwei Jobstarts schon vorkonfiguriert. Der eine bezieht sich auf die Funktionalität „Zentrales Journal“ und der andere auf die Funktionalität „Zentrale Administration“. Mehr Information über das Zentrale Journal erhalten Sie in einem separaten Benutzerhandbuch über das Zentrale Journal, wohingegen die Funktionalität „Zentrale Administration“ im Kapitel 13 dieses Handbuchs beschrieben ist.

### 3.3.1 Konfiguration der Jobstarts mittels GUI

GUI

Die Jobstarts werden im Administrationsfenster konfiguriert. Zu diesem Fenster gelangen Sie, indem Sie das Symbol Admin in der Funktionsleiste anklicken.

Wenn Sie sich im Admin-Fenster befinden, können Sie zwischen Jobstarts nach Empfang, Jobstarts nach Sendeversuch und Jobstarts nach Scheitern wählen, je nachdem, ob Sie eine bestimmte Aktion gekoppelt an den Empfang, den Versand von Dateien oder an das Scheitern eines Jobs auslösen möchten.

Einen neuen Jobstart können Sie anlegen, indem Sie im Admin-Fenster den Menüpunkt `Jobstart` wählen und anschließend - je nachdem unter welcher Voraussetzung der Job gestartet werden soll - Jobstart nach Empfang, Jobstart nach Sendeversuch oder Jobstart nach Scheitern markieren und mit der rechten Maustaste den Menüpunkt **Neuen Eintrag erstellen** wählen.

Nachdem Sie einen neuen Jobstarteintrag erstellt haben, bieten sich in der Jobstartmaske folgende Filter an:

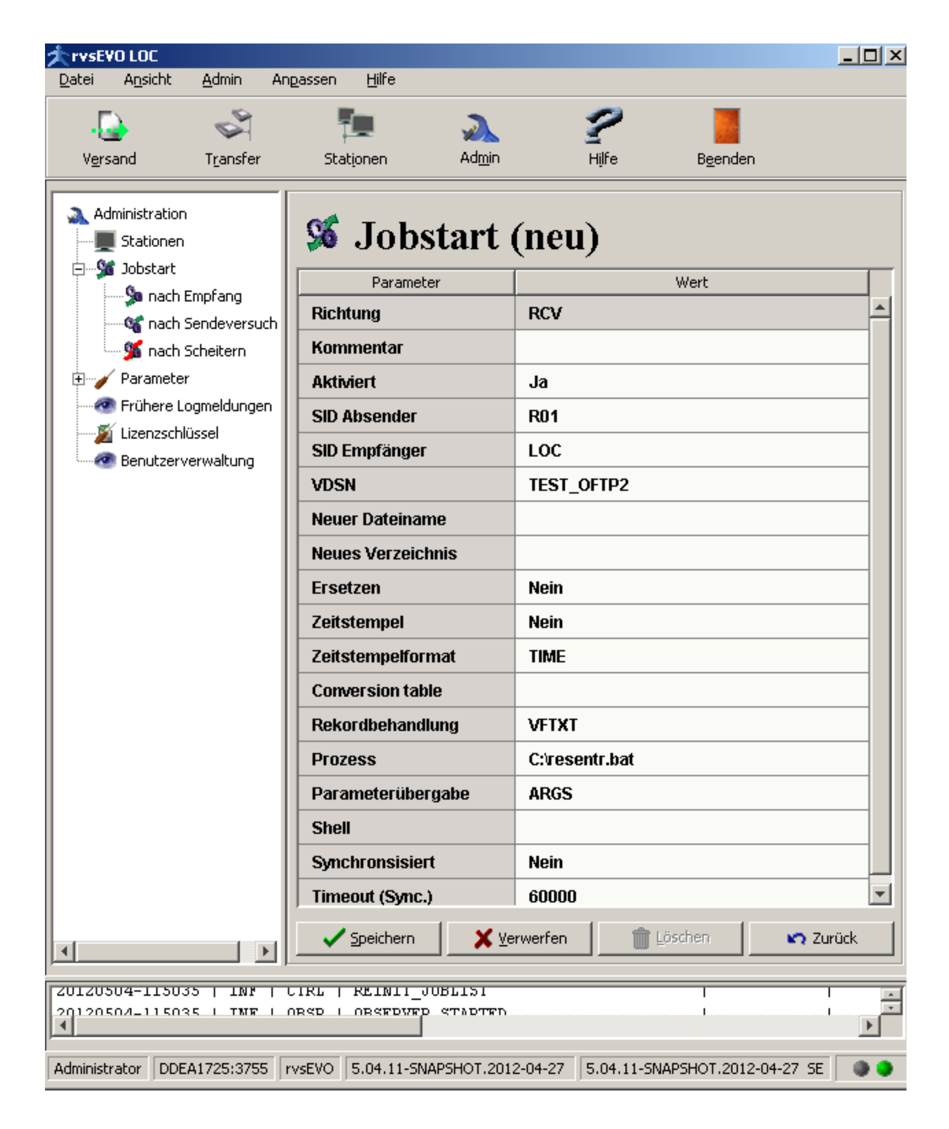
– alle Jobstarts: Kommentar, Aktiviert, SID Absender, SID Empfänger, VDSN, Prozess (Server), Prozess, Parameterübergabe, Shell, Synchronisiert und Timeout (Sync.)
– Jobstarts nach Empfang: Neuer Dateiname, Ersetzen, Zeitstempel, Zeitstempelformat, Conversion table und Rekordbehandlung

– Jobstarts nach Sendeversuch: Sendeversuche
Detailliertere Angaben zu den Filtern finden Sie in der Tabelle auf Seite 86 im nächsten Abschnitt.

GUI-Beispiele

**1. Beispiel: Jobstart nach Empfang**

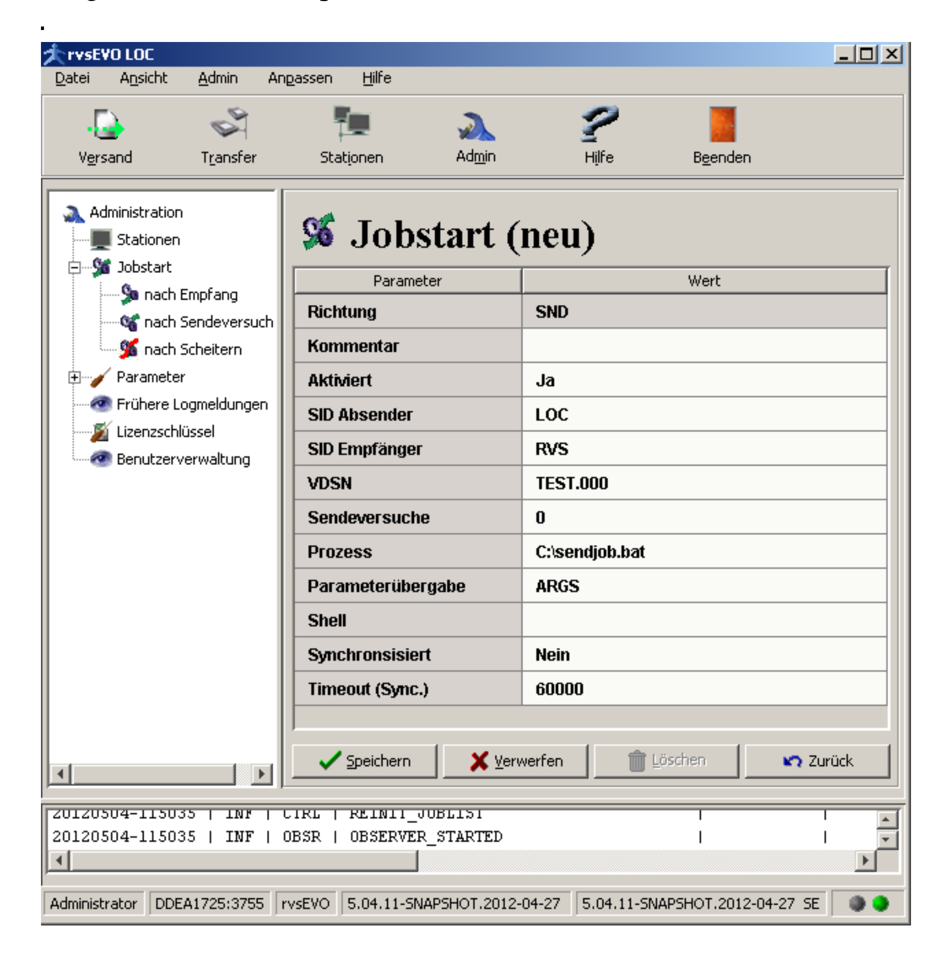
Beim Empfang der Datei mit dem virtuellen Dateinamen `TEST_OFTP2` von der gerouteten Station `R01` an die Station `LOC` startet rvsEVO das Programm `C:\resentr.bat`



**2. Beispiel: Jobstart nach Sendeversuch**

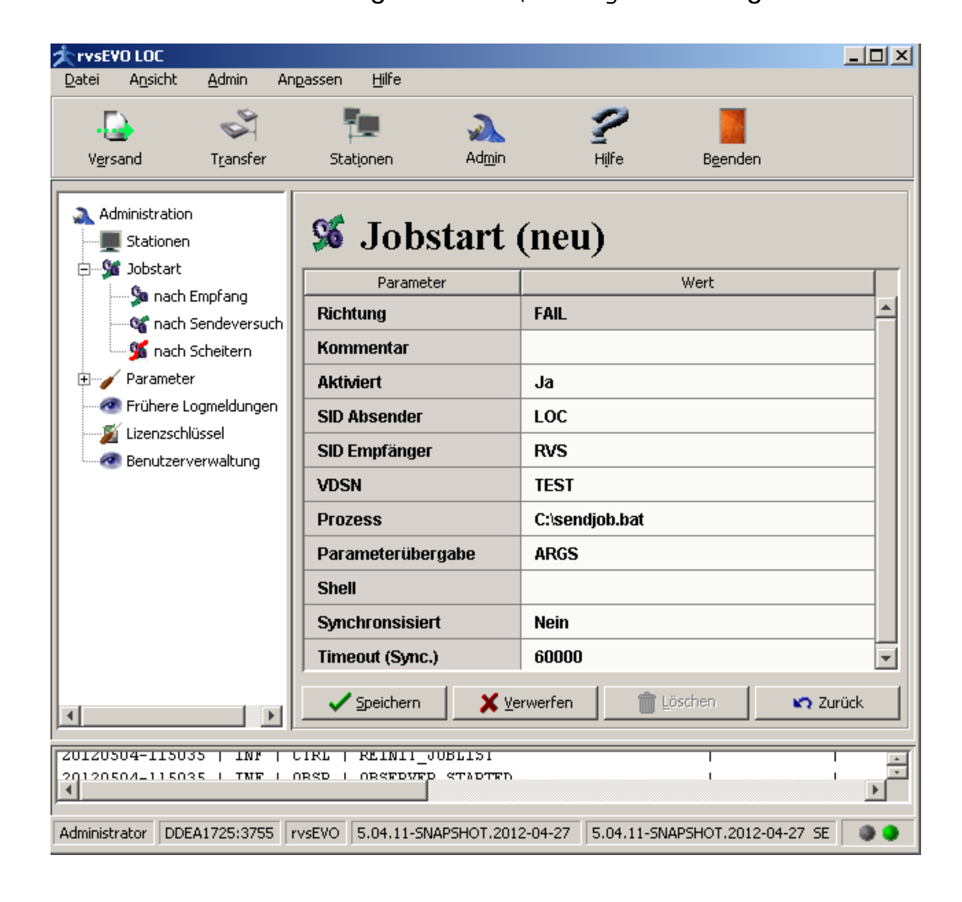
Wenn rvsEVO eine Datei mit dem Namen `TEST.000` an die Nachbarstation `RVS` erfolgreich (sendAttempts=0) versendet, startet das Programm `C:\sendjob.bat`

.



### 3. Beispiel: Jobstart nach Scheitern

Nach Erhalt einer Fehlermeldung beim Versand der Datei `TEST` an die Station `RVS` wird das Programm `C:\sendjob.bat` gestartet.




Eine Beschreibung der einzelnen Parameter finden Sie in der Tabelle im folgenden Kapitel.

**Hinweis**: Ein Jobstart nach Scheitern wird ausgelöst, wenn Jobs in Kategorie `FAILED` aufgenommen wurden oder wenn Jobs in der Kategorie `ACTIVE` in einen Fehlerstatus gegangen sind, der einen Operatoreingriff, wie z.B. Restart oder Delete, erfordert. Ebenso wird ein Jobstart nach Empfang eines NERP ausgelöst.

Folgende Situationen lösen einen Jobstart nach Scheitern aus:

Sendejob:

– Vorverarbeitung schlägt fehl (Status: `SP_FAILED`)

– Partnerstation ist nicht vorhanden

– Jobstart nach Sendeversuch schlägt fehlt (`ENDED_WITH_JS_ERROR`)

– SFNA (**S**tart **F**ile **N**egative **A**nswer) nach Verbindungsaufbau (z.B. Versand einer unverschlüsselten Datei an einen Empfänger, der Verschlüsselung fordert)

– `FATAL_ERROR` nach SFNA, z.B. wenn die Datei nicht vorhanden ist

– Jobstart nach erfolgreichem Versand schlägt fehlt (`ENDED_WITH_JS_ERROR`)
– EFNA-Fehler (**E**nd **F**ile **N**egative **A**nswer) nach Daten-Übertragung
– NERP-Empfang (Negative-End-to-End-Response)

Empfangsjob:

– Versand SFNA
– Versand EFNA (z.B. wenn der Byte Count der Datei beim Versand und beim Empfang nicht übereinstimmt.)
– Nachverarbeitung schlägt fehl, z.B. wenn der private Key zur Entschlüsselung fehlt (`SP_FAILED`)
– Jobstart bei Empfang schlägt fehl (`ENDED_WITH_JS_ERROR`)

### 3.3.2 Konfiguration der Jobstarts in der XML-Datei

Die XML-Konfigurationsdatei `rvsJobstart.xml` dient der Konfiguration der Jobstarts.

Diese Datei ist wie alle anderen rvsEVO - Konfigurationsdateien auch im XML-Format.

```
<jobstarterData>
       <jobfilters>
              <jobfilter>
                     <vdsn></vdsn>
                     <direction>SND</direction>
                     <sidOrig>LOC</sidOrig>
                     <sidDest>RVS</sidDest>
                     <sendAttempts>0</sendAttempts>
                     <process>C:\jobstart.bat</process>
                     <processingClass/>
                     <enabled>true<enabled>
                     <conversionTable>ANSI-IBM037</conversionTable>
              </jobfilter>
              <jobfilter>
               ...
              </jobfilter>
              ...
        </jobfilters>
</jobstarterData>
```

Diese Datei beinhaltet beliebig viele Jobfilter-Elemente. Die einzelnen Unterelemente von Jobfilter werden in der folgenden Tabelle genau beschrieben.

Unterelemente von Jobfilter

| **Parameter** | Beschreibung |
|---|---|
| **Aktiviert<br><enabled>** | Der Parameter legt fest, ob der Job gestartet werden soll. Mögliche Werte:<br>`true (Ja)`: Job starten (Standard)<br>`false (Nein)`: Job nicht starten |

| | |
|---|---|
| **Conversion Table <conversionTable>** | Für ASCII(ANSI) - EBCDIC Umwandlung stehen folgende Konvertierungstabellen zur Verfügung:<br>ASCII-IBM037, ASCII-IBM273, ANSI-IBM037, ANSI-IBM273.<br><br>Für EBCDIC - ASCII(ANSI) Umwandlung stehen folgende Konvertierungstabelle zur Verfügung:<br>IBM037-ASCII, IBM273-ASCII, IBM037-ANSI, IBM273-ANSI<br><br>Sie haben auch die Möglichkeit, eigene Konvertierungstabellen hinzuzufügen. Sehen Sie hierzu bitte den Hinweis zu Umwandlungstabellen auf Seite 92) |
| **Ersetzen <replace>** | Mit `Replace` legen Sie fest, wie mit einer empfangenen Datei verfahren wird, die im Namen mit einer lokalen Datei übereinstimmt<br>Mögliche Werte:<br>`true (Ja):` vorhandene Datei wird überschrieben<br>`false (Nein):` Datei wird nicht überschrieben, Zeitstempel wird an die neue Datei angehängt (Standard) |
| **Kommentar <comment>** | beliebiger Text |
| **Neuer Dateiname <newFilename>** | Mittels dieses Parameters können Sie einer empfangenen Datei einen neuen Namen/Pfad zuweisen. Sie haben folgende Möglichkeiten: Angabe des<br>... Dateinamens: Zustellung in Inbox mit neuem Namen (anstelle von VDSN).<br>... absoluter Pfad: Zustellung in angegebenes Verzeichnis (anstelle von Inbox) mit aktuellem Namen.<br>... absoluten Pfads und des Dateinamens: Zustellung in angegebenes Verzeichnis mit neuem Namen kein Eintrag: VDSN wird als Dateiname genommen (Standard).<br>Timestamp wird nach Timestamp-Regel angehängt. |

| | |
|---|---|
| **Parameterübergabe <parameterHandling>** | Mit diesem Parameter können Sie die Übergabe der Jobdaten an den Prozess einstellen.<br>Mögliche Werte:<br>`ARGS:` Jobdaten werden als Parameter / Argumente an das Skript übergeben (Standard)<br>`ENV:` Jobdaten werden als Umgebungsvariablen gesetzt.<br>`REPLACE:` Dient der Kompatibilität mit rvs® portable.<br>Für weitere Informationen lesen Sie bitte den Abschnitt "Parameterübergabe" auf Seite 90 |
| **Prozess (Server) <processingClass>** | Name des Parameters **Prozess** auf der Remote GUI. Die Beschreibung entnehmen Sie bitte dem Parameter **Prozess**. |
| **Prozess <process> (mandatory)** | Programm, das gestartet werden soll, wenn die Filterbedingungen alle zutreffen. Den Programmen wird ein fest definierter Satz von Parametern übergeben.<br>Parameter:<br>1 jobId<br>2 StationsID (des Senders bzw. des Empfängers)<br>3 Dateiname der gesendeten/bzw. empf. Datei<br>4 VDSN<br>5 Datum des Jobs<br>6 Uhrzeit des Jobs<br>7 Anzahl der Sendeversuche.<br>8 Format der Datei<br>9 Recordlänge<br>10 Übertragene Bytes<br>Lesen Sie bitte das Kapitel 4.5 "Versand einer Datei" für mehr Informationen über die Sendeparameter . Wie diese Parameter einzusetzen sind, entnehmen Sie dem Beispiel am Ende dieses Kapitels (Datei `$RVS_HOME/bin/jobstart/jobstart_detailed.bat`).<br>Nicht für Remote GUI. Bitte sehen Sie auch Parameter Prozess (Server) |
| **Rekordbehandlung <recordHandling>** | In diesem Parameter legen Sie fest, ob in empfangene Dateien im Format F oder V ein Zeilenumbruch eingefügt werden soll. Mögliche Werte:<br>`VFTXT`: Dateien im Format **F** oder **V** werden im Textmodus abgelegt. Das bedeutet, dass nach jedem Record einen Zeilenumbruch eingefügt wird (Standard).<br>`VTXT_FBIN`: Dateien im Format **V** werden im Textmodus und Dateien im Format **F** im Binärmodus abgelegt. |

| | |
|---|---|
| **Richtung**<br>**<direction>** | Richtung der Kommunikation (wird von rvsEVO vorgegeben, abhängig von dem Jobstarttypen). Mögliche Werte:<br>`SND` (bei Dateiversand), `RCV` (bei Dateiempfang) und `FAIL` (bei Jobverarbeitungsfehler). |
| **Sendeversuche**<br>**<sendAttempts>** | Anzahl der fehlgeschlagenen Sendeversuche. Wird hier "0" eingetragen, so bedeutet dies erfolgreicher Dateiversand. |
| **Shell**<br>**<shell>** | Nur für UNIX-Plattformen, wenn notwendig: Kommando-Shell, in der das Programm ausgeführt werden soll; z.B. `ksh`,.`csh`, .... |
| **SID Absender**<br>**<sidOrig>** | Stations-Id der Ursprungsstation (lokale oder virtuelle Station) |
| **SID Empfänger**<br>**<sidDest>** | Stations-Id der Zielstation |
| **Synchronisiert**<br>**<execSync>** | Bei Einstellung `true (Ja)`: OFTP-Session wird so lange aufrecht erhalten, bis das Programm abgearbeitet wurde. Standard: `false (Nein)` |
| **Timeout (Sync.)**<br>**<execTimeout>** | Bei Einstellung **Synchronisiert** = `Ja`: Abbruchzeit nach der die Verbindung abgebrochen wird. |
| **VDSN**<br>**<vdsn>** | Virtueller Dateiname (Regulärer Ausdruck als Filter), max. Länge 26 Zeichen |
| **Zeitstempel**<br>**<timestamp>** | Mit diesem Parameter legen Sie fest, nach welchen Regeln ein Zeitstempel generiert werden soll.<br>Mögliche Werte:<br>`true (Ja):` Zeitstempel wird immer angehängt<br>`false (Nein):` Zeitstempel wird nur bei Bedarf angehängt (Standard) |
| **Zeitstempelformat**<br>**<timestampFormat>** | Format des an den Dateinamen anzuhängenden Zeitstempels.<br>Mögliche Werte:<br>`TIME:` Uhrzeit der Erstellung des Jobs und fortlaufender Zähler im Format `hhmmssccc` (Standard)<br>`DATETIME:` Datum und Uhrzeit der Erstellung des Jobs und fortlaufender Zähler im Format `YYMMDD-hhmmssccc`.<br>`SFID_DATETIME:` Odette-Zeitstempel der Übertragung aus der SFID; Datum, Uhrzeit und fortlaufender Zähler im Format `YYYYMMDDhhmmsscccc`<br>`COUNTER:` 000000 bis 999999 (fortlaufender Zähler). Ist der Zähler (z.B. durch Löschen einer Datei) nicht fortlaufend, wird erst die Lücke gefüllt, bevor der Counter weiter hochgezählt wird. |

**Parameterübergabe**

**Hinweis**: Folgender Abschnitt bietet Ihnen eine genauere Beschreibung der Jobdaten-Übergabe an den Prozess bei den Einstellungen **parameterHandling**=ENV und **parameterHandling**=REPLACE

Einstellung **parameterHandling**=ENV: Jobdaten werden als Umgebungsvariablen gesetzt. Folgende Umgebungsvariablen sind definiert (in spitzen Klammern: Bezeichnung in XML-Datei, in Klammern: GUI-Name):

- **RVS_COMPRESSION** <compression> (Komprimierung)
- **RVS_CONVERSION_TABLE** <conversionTable> (Code-Tabelle)
- **RVS_CREATED_AT** <creationDate> (Erzeugt)
- **RVS_DATE:** Datum aus <creationDate> (Erzeugt)
- **RVS_DESCRIPTION:** <fileDescription> (Dateibeschr.)
- **RVS_DIRECTION:** <direction> (Richtung)
- **RVS_DISPOSITION:** <disposition> (Verbleib)
- **RVS_ENCRYPTION:** <encryption> (Verschlüsselung)
- **RVS_ENCRYPTION_ALGORITHM:** <encryptionAlgorithm> (Verschlüsselungsalgorithmus)
- **RVS_ERROR_ID:** <errorID> (Fehler-Nr.)
- **RVS_ERROR_TEXT:** <errorText> (Beschreibung)
- **RVS_EXTERNAL_JOBID**: <externalJobId> (Externe JobID)
- **RVS_FILENAME**: <filename> (Dateiname)
- **RVS_FILENAME_SRC**: <filenameSrc> (Original Dateiname)
- **RVS_JOB_ID**: <jobNumber> (ID)
- **RVS_LABEL:** <label> (Label)
- **RVS_LAST_CHANGE:** <lastStateChange> (letzte Änderung)
- **RVS_NERP_CREATOR_ODETTE_ID:** <nerpCreatorOdetteID> (Nerp Creator OdetteID)
- **RVS_NERP_REASON_CODE:** <nerpReasonCode> (Nerp Reason Code)
- **RVS_NERP_REASON_TEXT:** <nerpReasonText> (Nerp Reason Text)
- **RVS_RECORD_FORMAT**: <recordFormat> (Format)
- **RVS_RECORD_LENGTH**: <recordLength> (Satzlänge)
- **RVS_RESTART_POSITION:** <restartPos> (Restart Position)
- **RVS_SECURITY_FEATURE_SET:** <securityFeatureSet> (Sicherheitsmerkmale)
- **RVS_SEND_ATTEMPTS**: <sendAttempts> (Sendeversuche)
- **RVS_SERIALIZE:** <serialisation> (Serialisierung)
- **RVS_SID_DESTINATION:** <sidDestination> (SID Destination)
- **RVS_SID_ORIGINATOR:** <sidOriginator> (SID Originatior)
- **RVS_SIGNATURE:** <sign> (Dateisignatur)
- **RVS_SIGNATURE_EERP:** <signEERP> (signierten EERP/NERP beantragen)
- **RVS_STATUS:** <status> (Status)
- **RVS_TIME:** Uhrzeit aus <creationDate> (Erzeugt)
- **RVS_TIME_START_FILE**: <timeStartFile> (Time Start File)

– **RVS_TRANSMITTED_BYTES**: <transmittedBytes> (Übertragene Bytes)
– **RVS_VDSN**: <VDSN> (VDSN)

Die detailierte Beschreibung entnehmen Sie bitte der Tabelle im Abschnitt "Erläuterungen der Jobattribute:” auf Seite 133.

Einstellung **parameterHandling**=REPLACE: Die Einstellung REPLACE dient der Kompatibilität mit rvs® portable. Es wird eine Kopie der angegebenen Datei erzeugt. Dabei werden die Platzhalter in den Skripten durch die Jobdaten ersetzt. Die Platzhalter sind durch ? am Anfang und Ende begrenzt (z.B. wird "move ?DSN? /home/rvsevo/incomming" zu "move /home/rvsevo/rvsEVO/files/inbox/TEST.TXT /home/rvsevo/incomming")

Definierte Platzhalter für alle Jobstart-Types:

– ?**DSN**?: DataSetName (Name der lokalen Datei)
– ?**VDSN**?: virtueller Dateiname unter welchem die Datei übertragen wurde
– ?**DTAVAIL**?: Zeitpunkt zu dem die Datei für das Senden bereit war im Format TT/MM/JJ HH:MM:SS
– ?**FORMAT**?: Satzformat der Datei
– ?**BYTES?**: Anzahl der übertragenen Bytes
– ?**RECORDS**?: Zahl der übertragenen Sätze für Dateien im F und V Format; für Dateien im Format **T** und **U** immer 0 (null).
– ?**DTRCV**?: Zeitpunkt, zu dem die Datei dem lokalen Benutzer zugestellt wurde im Format TT.MM.JJ HH:MM:SS
– ?**UID**?: UserID
– ?**SID**?: StationsID
– ?**DSNTEMP**?: Name der temporären Datei

Definierte Platzhalter für alle Jobstart-Type SND:

– ?**MAXRECL**?: Die Bedeutung dieses Feldes hängt vom Satzformat der empfangenen Datei ab.
  – **F** Format: Länge jedes Satzes
  – **V** Format: maximale Länge, den ein Satz haben kann
  – **T** und **U** Format: Immer 0 (null)
– ?**LABEL**?: Zeichenfolge, wenn das Sendekommando ein **LABEL**-Parameter enthielt. Kann zur Identifikation des Sendekommandos benutzt werden.
– ?**SECN**?: rvsEVO-JobID (entspricht der Nummer des Sendeeintrags **SE** in rvs® portable)
– ?**SKCN**?: rvsEVO-JobID (entspricht der Nummer des Sendekommandos **SK** in rvs® portable)
– ?**SIDORIG**?: StationsID der Ursprungsstation
– ?**SENDATT**?: Anzahl der Sendeversuche, nach denen der Job gestartet werden soll.

Definierte Platzhalter für alle Jobstart-Type RCV:

- ?**MAXRECL**?: Die Bedeutung dieses Feldes hängt vom Satzformat der empfangenen Datei ab.
  - **F** Format: Länge jedes Satzes
  - **V** Format: maximale Länge, den ein Satz haben kann
  - **T** und **U** Format: Immer 0 (null)
- ?**CNQS**?: rvsEVO-JobID (entspricht der Kommandonummer der QuittungsSendung (**EERP**) für die empfangene Datei in rvs® portable ).
- ?**CNIE**?: rvsEVO-JobID (entspricht der Kommandonummer des Informations-Eingangs (IE) für die empfangene Datei in rvs® portable).
- ?**CNIZ**?: rvsEVO-JobID (entspricht der Kommandonummer der Informations-Zustellung (IZ) für die empfangene Datei in rvs® portable).

**Umwandlungs tabellen**

**Hinweis**: Folgender Absatz bietet Ihnen eine kurze Beschreibung der verschiedenen Codeumwandlungstabellen, die von rvsEVO unterstützt werden:

- **ASCII**: US-ASCII ISO 646; Bei diesem Zeichensatz handelt sich um einen 7-Bit-Code, d.h. es sind maximal 128 Zeichen (Codewerte 0 - 127) darstellbar. Dieser Zeichensatz ist eine Teilmenge vieler anderer Zeichensätze mit 256 Zeichen, unter ihnen auch des ANSI - Zeichensatzes für MS Windows.
- **ANSI**: Windows ANSI; Die Zeichen an den Stellen 0 bis 127 sind gleich wie beim ASCII-Zeichensatz und die meisten Zeichen an den Stellen 128 bis 255 sind gleich wie beim ISO-Latin 1-Zeichensatz.
- **EBCDIC 037**: unterstützt Zeichen aus den folgenden Ländern: Australien, Brasilien, Kanada, Neuseeland, Portugal, Südafrika, USA.
- **EBCDIC 273**: unterstützt Zeichen (insbesondere Umlaute) aus den folgenden Ländern: Deutschland, Österreich, Schweiz.

**Eigene Konvertierungstabellen hinzufügen:**

Wenn Sie eigene Umwandlungstabellen hinzufügen möchten, gehen Sie bitte wie folgt vor:

- Definieren Sie eine Konvertierungstabelle nach folgendem XML-Schema:

```
<conversiontable codein=Original-Characterset codeout=Ziel-Characterset>
    <description>Table to converts signs from ... to ...</description>
    <char value="0">0</char>
    <char value="1">1</char>
           .
           .
</conversiontable>
```

  Hierbei ist zu beachten, dass `char value` dem Ausgangswert entspricht und der zweite Wert in spitzen Klammern dem Zielcode. Eine detaillierte Beschreibung für die Erstellung von Konvertierungstabellen finden Sie auch in der Datei: `$RVS_HOME\conf\conversiontables\conversion-table-template.xml`
- Speichern Sie Ihre XML-Datei im Verzeichnis `$RVS_HOME\conf\conversiontables`

- Fügen Sie Ihre Tabelle zur Liste der Codeumwandlungstabellen hinzu, indem Sie die Datei `$RVS_HOME\conf\conversionta-bles\characterSetConverters.xml` editieren.

  Im Folgenden sehen Sie einen Auszug aus der Datei `characterSetConverters.xml`:

```
<characterSetConverters coding="">
    <!-- -->
    <converter cid="ANSI-IBM037">conversion_ANSI_IBM037.xml</converter>
   <converter cid="IBM273-ASCII">conversion_IBM273_ASCII.xml</
converter>
        .
        .
</characterSetConverters>
```

  In dem oben genannten Beispiel ist `cid="ANSI-IBM037"` (`cid="IBM273-ASCII"`) der Verwendungsname in rvsEVO und `conversion_ANSI_IBM037.xml` `(conversion_IBM273_ASCII.xml)` der dazugehörige Dateiname.

- Nach dem Hinzufügen neuer Konvertierungstabellen ist ein Neustart des rvs-Servers erforderlich.

**Jobstart nach fehlgeschlagenen Sendeversuchen**

Sollen Prozesse nach einer bestimmten Anzahl fehlgeschlagener Sendeversuche gestartet werden, wird bei dem Parameter **Sendeversuche** die Anzahl der Fehlvesuche angegeben, nach dem ein Job gestatet werden soll. Wenn mehrere Jobs mit dem gleichen Muster für Zielstation und dem gleichen virtuellen Dateinamen existieren, bei denen aber nach unterschiedlicher Anzahl von Sendeversuchen verschiedene Programme gestartet werden sollen, wird dieser Konflikt wie folgt aufgelöst: Es wird der Jobstart mit der höchsten Zahl bei **Sendeversuche,** durch die aktuelle Anzahl fehlgeschlagener Sendeversuche ohne Rest teilbar ist, ausgeführt.

**Beispiel**

**Beispiel**: Das Programm P1 soll nach einem Sendeversuch gestartet werden und das Programm P2 nach drei Sendeversuchen. Dann gilt:

1. Sendeversuch -> P1
2. Sendeversuch -> P1
3. Sendeversuch -> P2
4. Sendeversuch -> P1
5. Sendeversuch -> P1
6. Sendeversuch -> P2

**Löschen eines Jobstarteintrages**

Wenn Sie einen Jobstarteintrag löschen möchten, müssen Sie ihn zuerst mit Doppelklick auswählen. Im neuen Fenster haben Sie dann die

Möglicheit, den ausgewählten Jobstart mit der Schaltfläche **Löschen** zu entfernen.

**Skripte für Jobstartprozesse**

Jobstart-Skripte

Im Verzeichnis `$RVS_HOME\bin\jobstart` befinden sich die von rvsEVO mitgelieferten Beispiel-Skripte für Jobstartprozesse. Sie können diese Batch-Dateien an Ihre Bedürfnisse anpassen oder eigene Skripte erstellen. Im Folgenden finden Sie eine Kurzbeschreibung der von rvsEVO mitgelieferten Jobstart-Skripte:

**delaftersend.bat**

Dieses Skript kann in einem Jobstart nach Versand ausgeführt werden. Nach erfolgreichem Versand wird die versendete Datei gelöscht.

**handleMangement.bat**

Dieses Skript bezieht sich auf die Zentrale Administration. Weitere Informationen finden Sie im Kapitel 13 "Zentrale Administration von rvsEVO"

**jobstart.bat** und **jobstart_detailed.bat**

Diese Skripte sind in einem Jobstart nach Empfang und nach Versand ausführbar. Bei Ausführung dieser Skripte wird ein Protokoll, dass Informationen zur gesendeten Datei und die Transferdaten enthält, in die Datei `$RVS_HOME\log\jobstart.out` geschrieben. Bei `jobstart_detailed` erfolgt die Ausgabe formatiert, während bei `jobstart` die Daten unformatiert in `$RVS_HOME\log\jobstart.out` geschrieben werden.

**journal.bat**

Mit diesem Skript wird das Tool `sendJournal` gestartet, welches ein Journal erzeugt und dieses an das Zentrale Journal sendet.

Benötigter Parameter:

| | |
|---|---|
| **-f <filename>** | Name der Request-Datei des Zentralen Journals.<br>Default: ZJREQUEST |

Weitere Informationen finden Sie im Benutzerhandbuch „Zentrales Journal“.

**sendback.bat**

Beim Start dieses Skriptes wird die empfangene Datei an die Ursprungsstation zurückgesendet. Die Einstellung der Sendeparameter wird übernommen. Informationen zur Datei und zum Versand werden in der Datei `$RVS_HOME\log\jobstart.out` protokolliert.

# 4 Arbeiten mit rvsEVO

Batch-Dateien

In diesem Kapitel werden alle Programme, die Ihnen für den täglichen Umgang mit rvsEVO zur Verfügung stehen, beschrieben. Diese Programme befinden sich als Batch-Dateien (Windows) oder Skripte (UNIX) im Verzeichnis `$RVS_HOME\bin`.

**Hinweis**: Sie müssen sich im Verzeichnis `$RVS_HOME\bin` befinden, wenn Sie ein rvsEVO-Programm aufrufen.

Manche von den rvsEVO-Programmen (wie z.B. GUI, Start und Stopp von rvsEVO) stehen auch über die rvsEVO-Programmgruppe zur Verfügung (`Start -> Alle Programme -> rvsEVO`).

## 4.1 Starten des rvsEVO-Servers

`startServer`

Mit dem Programm `startServer` können Sie den rvsEVO-Server in der Eingabeaufforderung starten.

**Beispiel:**

```
startServer
```

Es können keine Parameter übergeben werden. Ein erfolgreicher Start sieht folgendermaßen aus:

```
*
* rvs Server has started.
*
```

**Hinweis**: Es ist auch möglich, den rvsEVO Server über die rvsEVO -Programmgruppe zu starten (`Start -> Programme -> rvsEVO -> rvsEVO GUI`). Dabei wird zuerst das Programm startGUI gestartet, welches seinerseits den rvsEVO-Server startet.

## 4.2 Stoppen des rvsEVO-Servers

`stopServer`

Stoppen Sie rvsEVO mit dem Programm `stopServer`.

**Syntax:**

```
stopServer -m <mode> [-verbose]
```

Alle Parameter sind optional:

| | |
|---|---|
| **-m <mode>** | Zeit für Beendigung der Jobs, bevor rvsEVO gestoppt wird.<br>Mögliche Werte:<br>`0` (default; 120 Sekunden),<br>`1` (60 Sekunden),<br>`2` (30 Sekunden),<br>`3` (20 Sekunden),<br>`4` (10 Sekunden). |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |

**Beispiel:**

```
stopServer
```

**Ergebnis**: Der Server wird in 120 Sekunden gestoppt.

**Beispiel:**

```
stopServer -m 3
```

**Ergebnis**: Der Server wird in 20 Sekunden gestoppt.

```
*
* rvs Server has stopped.
*
```

**Hinweis**: Es ist auch möglich, den rvsEVO Server über die rvsEVO - Programmgruppe zu stoppen (`Start -> Programme -> rvsEVO -> stop Server`).

## 4.3 Meldungen und Warnungen ausgeben

Die Meldungen und Warnungen von rvsEVO werden in den Log-Dateien im Verzeichnis `$RVS_HOME\log` gesichert.

Die Datei `monitor.log` beinhaltet Meldungen des Monitors, die Datei `rvs.log` Meldungen des rvsEVO-Servers und die Datei `rvsClient.log` Meldungen von der Eingabeaufforderung und der rvsEVO-Clients.

### 4.3.1 Meldungen des Monitors anzeigen

rvsEVO bietet im unteren Teil der Benutzeroberfläche die Möglichkeit, aktuelle LogMeldungen zu verfolgen.

Frühere Meldungen können Sie über die GUI oder mit dem Programm `showMonitorLog` anzeigen lassen.

`monitor.log`

Die Melungen des Monitors werden in die Datei `monitor.log` geschrieben. Es wird täglich eine neue `monitor.log`-Datei generiert. Der Name der vorigen `montor.log`-Datei wird um Datepattern und fortlaufenden Zähler erweitert. Ebenso wird bei Erreichen der maximalen Dateigröße eine neue `monitor.log`-Datei generiert. Die Datei mit dem höchsten Counter ist die zuletzt generierte.

In der XML-Datei `$RVS_HOME\conf\rvsLogger` können Sie im XML-Element `appender name="monlog"` die maximale Größe und die maximale Anzahl der Log-Dateien bestimmen:

– Maximale Größe: Bei erreichen der im Parameter `MaxFileSize` angegebenen Größe wird eine neue Log-Datei generiert.

  Syntax:

  ```
  <param name="MaxFileSize" value="10MB"/>
  ```

– Anzahl der Log-Dateien: Der Wert des Parameters `MaxRollFileCount` bestimmt die Anzahl der Log-Dateien, die im Ordner `$RVS_HOME\log` generiert werden können.

  Syntax:

  ```
  <param name="MaxRollFileCount" value="50"/>
  ```

**GUI**

**LogMeldungen**

Beim Start der GUI werden standardmäßig die letzten 25 LogMeldungen im unteren Bereich des Fensters angezeigt. Diese Einstellung ist in der Datei `$RVS_HOME\conf\rvs-system.properties` konfigurierbar, indem der Wert des Parameters `rvs_evo.monlog.initial_buffer_size` gesetzt wird. Die Raute (#) zu Beginn der Zeile muss gelöscht werden, da die Änderung andern-

falls ignoriert wird. In folgendem Beispiel werden nach einem Neustart von rvsEVO die letzten 50 LogMeldungen angezeigt.

**Beispiel:**

```
rvs_evo.monlog.initial_buffer_size=50
```

Die Darstellung der LogMeldungen wird in der Datei `rvs-system.properties` durch die Definition `rvs_evo.monitorlog.client_layout` gesteuert.

**Beispiel:**

```
rvs_evo.monitorlog.client_layout=%1$-11s * %3$-4s *
%4$-36s * %8$-s
```

In obigem Beispiel enspricht `1$` dem Ausgabe-Parameter, `-` bedeutet linksbündig (Standard: rechtsbündig), `11` ist die Breite der Spalte (11 Zeichen) und als Trennungszeichen ( `*` ) kann beliebiger Text gewählt werden. Jeder Ausdruck wird durch `%` am Anfang und `s` am Ende begrenzt. Beim letzten Parameter entfällt die Längenangabe.

Folgende Informationen können angezeigt werden:

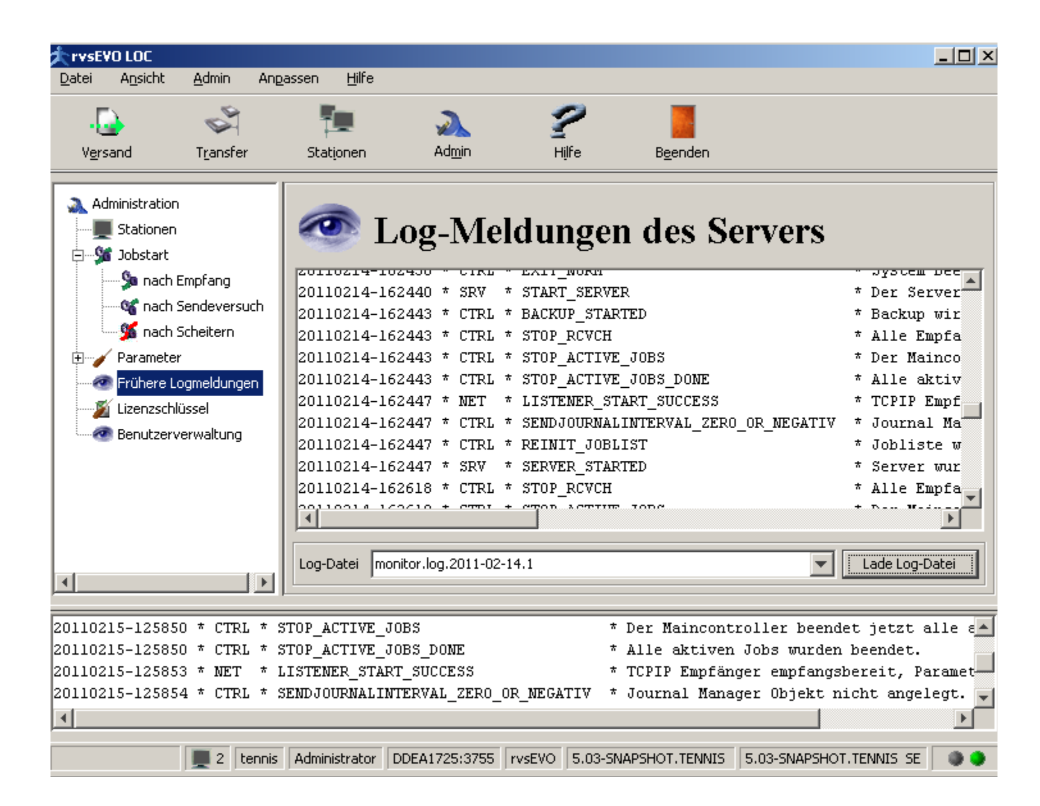
| Parameter | Beschreibung |
|---|---|
| `1$` **(time)** | Datum und Uhrzeit des Eintrags im Format `yyyyMMdd-HHmmss` |
| `2$` **(level)** | Art der Meldung. Mögliche Werte:<br>`INF`: Information<br>`WRN`: Warning / wichtiger Hinweis<br>`ERR`: ERROR |
| `3$` **(module)** | rvsEVO-Modul, welches die Meldung sendet. Mögliche Werte:<br>SRV: rvsEVO-Server<br>`CTRL`: Controller<br>`NET`: Netzwerk<br>`OFTP`: Odette-Protokoll<br>`CONF`: Konfigration<br>`PERS`: Persistenz<br>`ACX`: automatischer Zertifikatsaustausch<br>`SP`: Service Provider<br>`OBSR`: Observer |
| `4$` **(messageKey)** | Nachrichtenschlüssel |
| `5$` **(sessionID)** | ID der OFTP-Session |
| `6$` **(stationID)** | SID der Nachbarstation |
| `7$` **(jobID)** | rvsEVO-JobID |

| | |
|---|---|
| 8$ **(localized message)** | Textnachricht aus Nachrichtenschlüssel (GUI) oder Parameter für die Textnachricht abgeleitet aus dem Nachrichtenschlüssel (Log-Datei) |

**Hinweis:** Ab rvsEVO-Version 5.04 können die LogMeldungen bei Verwendung einer Oracle-Datenbank zusätzlich im Oracle-Benutzerbereich in der Tabelle MONITOR_MESSAGE gesichert werden. Diese Funktionalität ist in der XML-Datei $RVS_HOME\conf\rvsLogger konfigurierbar. Bitte aktivieren Sie hierzu die Definitionen <appender-ref ref="DB"/> und <appender name="DB" class="com.gedas.rvs.evo.log.db.MonitorLogDBAppender"> indem Sie die Zeichenfolge <!-- zu Beginn des Satzes und --> am Ende des Satzes löschen.

**Frühere LogMeldungen**

Die früheren Meldungen des Monitors können im Admin-Fenster im Baum Administration, Menüpunkt **Frühere LogMeldungen** angezeigt werden. Zum Admin-Fenster gelangen Sie, indem Sie das Symbol Admin in der Funktionsleiste anklicken. Bitte wählen Sie eine Log-Datei aus und betätigen Sie anschließend die Schaltfläche **Lade Log-Datei**, um frühere LogMeldungen einzusehen.




**Eingabeaufforderung (Kommandozeile)**

show MonitorLog

Benutzen Sie das Programm showMonitorLog um die Meldungen des Monitors zu verfolgen und eventuell Fehlermeldungen zu analysieren.

**Syntax:**

`showMonitorLog`

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

Optionale  Parameter:

| | |
|---|---|
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-help** | Gibt Beschreibung zum aktuellen Kommando aus |
| **-?** | Fordert Hilfe an. |

**Beispiel**:

```
C:\WINDOWS\system32\cmd.exe - showMonitorLog

C:\rvsEVO\bin>showMonitorLog

Ready to receive monitor log lines from server.

20110215-125850 * CTRL * BACKUP_STARTED                          * Backup wird gest
artet.
20110215-125850 * CTRL * STOP_RCVCH                              * Alle Empfangskan
õle werden gestoppt.
20110215-125850 * CTRL * STOP_ACTIVE_JOBS                        * Der Maincontroll
er beendet jetzt alle aktiven Jobs.
20110215-125850 * CTRL * STOP_ACTIVE_JOBS_DONE                   * Alle aktiven Job
s wurden beendet.
20110215-125853 * NET  * LISTENER_START_SUCCESS                  * TCPIP Empfõnger
empfangsbereit, Parameter: DDEA1725:3305.
20110215-125854 * CTRL * SENDJOURNALINTERVAL_ZERO_OR_NEGATIV     * Journal Manager
Objekt nicht angelegt.
20110215-125854 * CTRL * REINIT_JOBLIST                          * Jobliste wird re
initialisiert.
20110215-125854 * SRV  * SERVER_STARTED                          * Server wurde erf
olgreich gestartet.
20110215-134207 * CTRL * OUTGOING_CALL_INITIATED                 * Anruf zu TENNISE
 eingeleitet.
20110215-134207 * NET  * CONNECTION_CREATE_SUCCESS               * TCPIP Verbindung
 aufgebaut, Parameter: DDEA1725:3307.
```

`showMonitorLog-File` Eine ähnliche Funktionalität wie das Programm `showMonitorLog` bietet das Programm `showMonitorLogFile`. Damit kann eine Monitor-Log-Datei im Text-Modus betrachtet werden.

**Syntax:**

`showMonitorLogFile [-i <filename>]`

Pflicht-Parameter:

| **Parameter** | Beschreibung |
|---|---|
| **-i** | Pfad und Name der Monitor-Log-Datei; es muss ein existierender Dateiname sein. |

Optionale Parameter:

| | |
|---|---|
| **-?** | Fordert Hilfe an. |
| **-help** | Gibt Beschreibung zum aktuellen Kommando aus |

### 4.3.2 Meldungen des rvsEVO-Servers

In der Datei `$RVS_HOME\log\rvs.log` werden Fehlermeldungen des rvsEVO-Servers gesichert. Diese können Sie sich nicht über die GUI anzeigen lassen, sondern nur direkt einsehen.

In der XML-Datei `$RVS_HOME\conf\rvsLogger` können Sie im XML-Element `appender name="rvslog"` die maximale Größe und die maximale Anzahl der Log-Dateien bestimmen:

– Maximale Größe: Bei erreichen der im Parameter `maxFileSize` angegebenen Größe wird eine neue Log-Datei generiert.

  Syntax:

```
<param name="maxFileSize" value="2097152"/>
```

– Anzahl der Log-Dateien: Der Wert des Parameters `maxBackupIndex` bestimmt die Anzahl der Log-Dateien, die im Ordner `$RVS_HOME\log` generiert werden können.

  Syntax:

```
<param name="maxBackupIndex" value="10"/>
```

### 4.3.3 Meldungen der Eingabeaufforderung und der rvsEVO-Clients

In der Datei `$RVS_HOME\log\rvsClient.log` werden Warnungen und Fehlermeldungen von der Eingabeaufforderung und von den rvsEVO-Clients gesichert.

In der XML-Datei `$RVS_HOME\conf\rvsClientLogger` können Sie im XML-Element `appender name="Standard"` die maximale Größe und die maximale Anzahl der Log-Dateien bestimmen:

– Maximale Größe: Bei erreichen der im Parameter `maxFileSize` angegebenen Größe wird eine neue Log-Datei generiert.

  Syntax:

```
<param name="maxFileSize" value="2097152"/>
```

– Anzahl der Log-Dateien: Der Wert des Parameters `maxBackupIndex` bestimmt die Anzahl der Log-Dateien, die im Ordner `$RVS_HOME\log` generiert werden können.

  Syntax:

```
<param name="maxBackupIndex" value="10"/>
```

## 4.4 Aktivieren einer Station

Um Dateien abzuholen, die für Ihre Station bestimmt sind, können Sie eine direkte Nachbarstation mittels GUI oder mittels der Eingabeaufforderung aktivieren.

GUI

Um eine Nachbarstation über die GUI zu aktivieren, öffnen Sie das Stationsfenster in der Funktionsleiste, markieren die entsprechende Station und betätigen die rechte Maustaste. Wählen Sie aus dem Kontextmenü den Eintrag **Verbindung aktivieren** aus, um die Verbindung zu aktivieren.

`activate Station`

Mit dem Programm `activateStation` können Sie eine direkte Nachbarstation aktivieren, um die Dateien, die für Ihre Station bestimmt sind, abzuholen (siehe auch Kapitel 3.2.3 "Nachbarknoten hinzufügen und konfigurieren").

**Syntax:**

```
activateStation [-verbose][-help]
[-su <user> -sp <password> -sh <host:port>]
```

Optionale Parameter:

| | |
|---|---|
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-help** | Gibt Beschreibung zum aktuellen Kommando aus. |
| **-?** | Fordert Hilfe an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

Wenn die Aktivierung der Nachbarstation nicht erfolgreich sein sollte (z.B. wegen einer falschen IP-Adresse), müssen Sie auf die Nachrichten in dem Eingabeaufforderungsfenster, die mit **Meldung** oder **Error** beginnen, achten.

Im Falle einer falschen IP-Adresse, muss die Stationskonfiguration korrigiert werden.

## 4.5 Versand einer Datei

Der Versand einer Datei ist mittels der GUI und mittels der Eingabeaufforderung möglich.

### GUI

Zum Fenster **Sendeauftrag erstellen** gelangen Sie, indem Sie in der Funktionsleiste das Symbol **Transfer** und anschließend im Transfer-Baum den Menüpunkt **Sendeauftrag erstellen** anklicken oder direkt über das Symbol **Versand** in der Funktionsleiste.

Im Fenster **Sendeauftrag erstellen** können Sie Ihre Daten für den aktuellen Sendeauftrag eintragen oder auswählen  und durch betätigen der Schaltfäche **Senden** den Sendeauftrag starten.

Über die Schaltfläche **Als Standard speichern** bietet rvsEVO die Möglichkeit, Ihre aktuelle Einstellung zu speichern und zu einem späteren Zeitpunkt durch Anklicken der Schaltfläche **Standardwerte** wiederherzustellen. Über die Schaltfläche **Zurücksetzen** können Sie die Ursprungseinstellung wiederherstellen.




Sendeparameter

Pflicht-Parameter:

| | |
|---|---|
| **Datei** | Dateiname der zu versendenden Datei |
| **SID Empfänger** | StationsID des Empfängers |

| | |
|---|---|
| **VDSN** | Virtueller Dateiname; das ist der Dateiname, der für die ODETTE-Übertragung benutzt wird, max. 26 Zeichen |

Optionale Parameter:

| | |
|---|---|
| **Ausführungstermin** | Zeitpunkt, zu dem der Sendeauftrag ausgeführt werden soll. Eingabe im Format<br><`yyyy-mm-dd hh:mm:ss`> Datum und Uhrzeit<br><`yyyy-mm-dd`> nur Datum: der Sendeauftrag wird am angegebenen Datum um 00:00 Uhr ausgeführt<br><`hh:mm:ss`> Sendeauftrag wird am selben oder nächsten Tag zur angegebenen Zeit ausgeführt<br>Liegt der Zeitpunkt in der Vergangenheit, wird der Sendeauftrag sofort ausgeführt.<br><br>Bei der Eingabe des Wertes “h“ wird der Job im Status `HOLD` erzeugt. |
| **Datei-Signatur** | Dieser Parameter wird nur im Zusammenhang mit dem Sicherheitsmechanismus 4 / OFTP 2.0 (CMS) aktiviert. Mögliche Werte:<br><br>– J (ja): Datei wird signiert.<br>– N (nein): Datei wird nicht signiert. |
| **Externe JobID** | Parameter für die Angabe einer externen JobID. Eine externe JobID kann sich auf mehrere rvsEVO-JobIDs beziehen.Sie ist freiwählbar und besteht aus alphanumerischen Zeichen. |
| **File description** | Textkommentar, möglich erst ab der OFTP Version 2. Wenn Ihre Gegenstelle kein OFTP 2.0 kann, wird dieses Feld ignoriert. |
| **Format** | Format der zu übertragenden Datei:<br><br>– T (Text): eine Folge von ASCII-Zeichen<br>– F (Fest): feste Satzlänge<br>– V (Variabel): variable Satzlänge<br>U (Unstrukuriert): binäre Datei.<br><br>Siehe auch Paramter **Satzmodus**. |

| | |
|---|---|
| **Konvertierungstabelle** | Für ASCII(ANSI) - EBCDIC Umwandlung stehen folgende Konvertierungstabellen zur Verfügung:<br>ASCII-IBM037, ASCII-IBM273, ANSI-IBM037, ANSI-IBM273.<br><br>Für EBCDIC - ASCII(ANSI) Umwandlung stehen folgende Konvertierungstabelle zur Verfügung:<br>IBM037-ASCII, IBM273-ASCII, IBM037-ANSI, IBM273-ANSI<br><br>Sie haben auch die Möglichkeit, eigene Konvertierungstabellen hinzuzufügen. Sehen Sie hierzu bitte den Hinweis zu Umwandlungstabellen auf Seite 92) |
| **Label** | Alle Dateien, die in der gleichen Gruppe versendet werden, müssen die gleiche Kennung (Label) haben. Siehe auch Parameter **Serialisierung** |
| **Max.Satzlänge** | Für Dateien im Format Fest geben Sie die feste Satzlänge an, mit der der Empfänger die Datei interpretieren soll. Dies ist die Satzlänge jedes Satzes bis zum Zeilenwechsel (CR/LF bei MS Windows, LF bei UNIX-Systemen). Für Dateien im Format Variabel geben Sie die maximale Satzlänge an. Siehe auch Parameter **Satzmodus**. |
| **Offline Komprimierung** | Odette-Komprimierung während der Übertragung. Mögliche Werte: **Y**(Yes)/**N**(No).<br>Standard: **N** |
| **Ohne Kopie** | Beim Erzeugen eines Sendeeintrags wird standardmäßig die zu versendende Datei ins Outbox-Verzeichnis von rvsEVO kopiert und von hier versendet. Mit dieser Option kann dies unterbunden werden. In diesem Falle wird die Originaldatei nach erfolgreichem Versand gelöscht. Werte:<br>**Ja**: Datei wird nicht ins Outbox-Verzeichnis kopiert<br>**Nein**: Datei wird ins Outbox-Verzeichnis kopiert (Standard)<br>**Achtung!** Diese Option hat jedoch keine Auswirkung, wenn die Datei im Format `U` mit Kodekonvertierung versendet wird! |

| | |
|---|---|
| **Satzmodus** | In diesem Parameter können Sie für Dateien im Format 'F' oder 'V' angeben, ob es sich um eine Textdatei oder um eine Binärdatei handelt. Mögliche Werte:<br><br>**TXT**: Textdatei<br>**BIN**: Binäre Datei<br><br>Bei Textdateien im Format ’F’ werden Zeilen, die länger sind als im Parameter **Max.Satzlänge** vorgegeben, auf die definierte Länge abgeschnitten, kürzere Zeilen werden mit Leerzeichen aufgefüllt. Bei Textdateien im Format ’V’ werden längere Zeilen abgeschnitten, kürzere bleiben unverändert.<br>Bei binären Dateien haben alle Sätze außer dem letzten die maximale Satzlänge. Ist der letzte Satz kürzer als in **Max.Satzlänge** definiert, wird dieser bei Dateien im Format ’F’ mit binären Nullen aufgefüllt. |
| **Serialisierung** | **Y**(Yes)/**N**(No) Diese Option gewährleistet, dass Ihre Dateien in einer geordneten Reihenfolge ankommen. Alle Dateien, die in der gleichen Gruppe versendet werden sollen, müssen die gleiche Kennung (Label) haben. In der GUI dient VDSN als Label. |

| | |
|---|---|
| **Sicherheitsmerkmale** | Dieser Parameter bezieht sich auf die Art der Dateiverschlüsselung. Je nachdem welche Art ausgewählt wird, werden die Parameter, die sich auf die jeweilige Verschlüsselungsart beziehen, aktiviert.<br><br>Folgende Werte sind möglich:<br><br>– 1 / ohne: keine Verschlüsselung ist erwünscht<br>– 2 / ComSecure (V1)<br>– 3 / ComSecure (V2)<br>– 4 / OFTP 2.0 (CMS)<br><br>rvsEVO kann in zwei verschiedenen Formaten Dateien verschlüsseln: ComSecure (ein eigenes Fomat) und CMS (basiert auf X.509-Zertifikaten).<br><br>CMS ist nur im Zusammenhang mit OFTP Version 2 möglich (auch die Gegenstelle muss OFTP Version 2.0 können).<br><br>Bei dem Format ComSecure wird zwischen der Version 1 (Produktversionen 1.1 und 1.2) und 2 (Produktversion 1.3) unterschieden. Version 2 wird verlangt, wenn Dateien größer als 4 Gbyte vor der Übertragung verschlüsselt werden sollen. Die Gegenstelle muss auch mit ComSecure Version 2 verschlüsseln können. Wenn es sich um kleinere Dateien handelt und die Gegenstelle das Format Com-Secure Version 1 verwendet, ist 2 /ComSecure (V1) ausreichend. |
| **SID Absender** | StationsID des Absenders: lokale Station (Standard) oder virtuelle Station |

| | |
|---|---|
| **signierten EERP/ NERP (Quittungssi-gnatur) beantragen** | Nur im Zusammenhang mit dem Sicherheits-mechanismus 4 /OFTP 2.0 (CMS). Laut OFTP 2.0 haben Sie die Möglichkeit die Signatur der Quittung (EERP oder NERP) von Ihrem Partner zu beantragen. Allerdings muss der Partner auch die OFTP Version 2 unterstützen. Im Falle, dass Sie eine signierte Quittung beantragen und Ihr Partner (Destination - Endstation) sie Ihnen nicht sendet, endet der rvsEVO-Sendjob in den Status `FAILED` (fehlgeschlagen). Mögliche Werte:<br>– `J` (ja)<br>– `N` (nein). |
| **Verbleib** | Dieser Parameter entscheidet, ob die zu versendende Datei nach dem Versand bei Ihnen lokal gelöscht wird oder erhalten bleibt. Mögliche Werte:<br>– **K** (Keep): Datei bleibt nach dem Versand erhalten.<br>– **D** (Delete): Datei wird nach dem Versand gelöscht. |
| **Verschlüsselung** | Legt fest, ob Dateiverschlüsselung beim Dateitransfer angewandt wird. Mögliche Werte: **Y**(Yes)/**N**(No).<br>Standard: **N**<br>Mehr über Dateiverschlüsselung lesen Sie bitte im Kapitel 6. |
| **Verschlüsselungsal-gorithmus** | Dieser Parameter steht nur dann zur Verfü-gung, wenn als Sicherheitsmechanismus 4 / OFTP 2.0 (CMS) ausgewählt wurde. Folgende Algorithmen stehen zur Auswahl:<br>– Kein<br>– DES_EDE3_CBC (Triple DES)<br>– AES256_CBC |

### Eingabeaufforderung

createSendJob

Mit dem Programm `createSendJob` erstellen Sie einen Sendeauftrag.

**Syntax:**

```
createSendJob -d <filename> -s <receiver sid>
-v <vdsn>
[-t <table name>] [-F <format>] [-M <length>]
[-S <serialize> -l <label>] [-nocp]
[-D <disposition>] [-C] [-Y] [-rsr] [-sfs <set-id>]
[-sif] [-Yalg <encryption algorithmen>]
[-Ycin <issuer name>] [-Ycsn <serial no>]
[-sifcin <issuer name>] [-Ycsn <serial no>]
[-help] [-?] [-verbose] [-Sd <init date>]
[-St <init time>] -[Sh] [-idf <filename2>]
[-xid <external job id>] [-so <originator sid>] [-rm
<record mode>]
[-su <user> -sp <password> -sh <host:port>]
```

Benötigte Parameter:

| | |
|---|---|
| **-d <filename>** | Dateiname der zu versendenden Datei. |
| **-s <receiver sid>** | StationsID des Empfänger. |
| **-v <vdsn>** | Virtueller Dateiname; das ist der Dateiname, der für die ODETTE-Übertragung benutzt wird, max. 26 Zeichen |

Optionale Sende-Parameter:

| | |
|---|---|
| **-?** | Fordert Hilfe (Usage) an. |
| **-C** | ODETTE-Komprimierung (Offline-Komprimierung). |
| **-D <disposition>** | Diese Option legt fest, ob die versendende Datei nach erfolgreichem Versand gelöscht werden soll. Mögliche Werte: **K** (keep) - Datei wird nicht gelöscht, Standard; **D** (delete) - Datei wird gelöscht. |
| **-desc <description>** | Dateibeschreibung, möglich nur mit **sfs=4** (OFTP 2.0) |
| **-F <format>** | Format der zu übertragenden Datei:<br>– T (Text): eine Folge von ASCII-Zeichen<br>– F (Fest): feste Satzlänge<br>– V (Variabel): variable Satzlänge<br>– U (Unstrukuriert): binäre Datei.<br>Siehe auch Parameter **-rm** (Satzmodus). |
| **-h** | Fordert Hilfe (Usage) an. |

| | |
|---|---|
| **-idf <filename>** | Die rvsEVO-JobID des erzeugten Jobs wird bei erfolgreichem Versand (return code/error level = 0) in eine vorgegebene Datei (filename) geschrieben. |
| **-j <start job>** | zur Zeit ohne Funktion |
| **-l <label>** | Alle Dateien, die in der gleichen Gruppe versendet werden, müssen die gleiche Kennung (Label) haben. Siehe auch Option `-s <serialize>` |
| **-M <length>** | Für Dateien im Format Fest geben Sie die feste Satzlänge an, mit der der Empfänger die Datei interpretieren soll. Dies ist die Satzlänge jedes Satzes bis zum Zeilenwechsel (CR/LF bei MS Windows, LF bei UNIX-Systemen). Für Dateien im Format Variabel geben Sie die maximale Satzlänge an. Siehe auch Parameter **-rm** (Satzmodus). . |
| **-nocp** | Beim Erzeugen eines Sendeeintrags wird standardmäßig die zu versendende Datei ins Outbox-Verzeichnis von rvsEVO kopiert und von hier versendet. Mit der Option `-nocp` kann dies unterbunden werden. In diesem Falle wird die Originaldatei nach erfolgreichem Versand gelöscht. **Achtung!** Diese Option hat jedoch keine Auswirkung, wenn die Datei im Format `U` mit Kodekonvertierung versendet wird! |
| **-rm <record mode>** | In diesem Parameter können Sie für Dateien im Format 'F' oder 'V' angeben, ob es sich um eine Textdatei oder um eine Binärdatei handelt. Mögliche Werte:<br>**TXT**: Textdatei<br>**BIN**: Binäre Datei<br>Bei Textdateien im Format ’F’ werden Zeilen, die länger sind als im Parameter **Max.Satzlänge** vorgegeben, auf die definierte Länge abgeschnitten, kürzere Zeilen werden mit Leerzeichen aufgefüllt. Bei Textdateien im Format ’V’ werden längere Zeilen abgeschnitten, kürzere bleiben unverändert.<br>Bei binären Dateien haben alle Sätze außer dem letzten die maximale Satzlänge. Ist der letzte Satz kürzer als in **Max.Satzlänge** definiert, wird dieser bei Dateien im Format ’F’ mit binären Nullen aufgefüllt. |

| | |
|---|---|
| **-rsr** | Signierter EERP/NERP (Quittungssignatur) wird beantragt. Nur mit **sfs=4** (OFTP 2.0). |
| **-S <serialize>** | **Y**(Yes)/**N**(No). Diese Option gewährleistet, dass Ihre Dateien in einer geordneten Reihenfolge ankommen. Alle Dateien, die in der gleichen Gruppe versendet werden sollen, müssen die gleiche Kennung (Label) haben. In der GUI dient VDSN als Label und hier (in der Eingabeaufforderung) kann Label (siehe Parameter Label) angegeben werden. |
| **-Sd <init date>** | Datum, an dem der Sendeauftrag ausgeführt werden soll. Eingabe im Format `<yyyy-mm-dd>` |
| **-sfs <set-id>** | Entspricht dem Parameter **Sicherheitsmerkmale** in der GUI.<br>Mögliche Werte:<br>– 1 (None)<br>– 2 (ComSecure V1)<br>– 3 (ComSecure V2<br>– 4 (OFTP 2.0) |
| **-Sh** | Sendeauftrag wird im Status `HOLD` erzeugt. (Freigabe: Siehe Kapitel 4.10.1 "Löschen, Anhalten, Freigabe von EERPs und aktiven Sendeübertragungen"). |
| **-sif** | Dateisignatur wird aktiviert. Nur mit **sfs=4** (OFTP 2.0). |
| **-sifcsn <serial no>** | Serielle Nummer des Zertifikats für Dateisignatur |
| **-sifcin <issuer name>** | Herausgebername des Zertifikats für Dateisignatur |
| **-so <originator sid>** | StationsID des Versenders: lokale (Standard) oder virtuelle Station |
| **-St <init time>** | Zeitpunkt, zu dem der Sendeauftrag ausgeführt werden soll. Eingabe im Format `<hh:mm:ss>` |

| | |
|---|---|
| **-t <table name>** | Name Ihrer Codeumwandlungstabelle.<br>Mögliche Werte:<br>Für ASCII(ANSI) - EBCDIC Umwandlung: ASCII-IBM037, ASCII-IBM273, ANSI-IBM037, ANSI-IBM273.<br>Für EBCDIC - ASCII(ANSI) Umwandlung: IBM037-ASCII, IBM273-ASCII, IBM037-ANSI, IBM273-ANSI<br><br>Sie haben auch die Möglichkeit, eigene Konvertierungstabellen hinzuzufügen. Sehen Sie hierzu bitte den Hinweis zu Umwandlungstabellen auf Seite 92) |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-xid <external job id>** | Parameter für die Angabe einer externen JobID. Eine externe JobID kann sich auf mehrere rvsEVO-JobIDs beziehen.Sie ist freiwählbar und besteht aus alphanumerischen Zeichen. |
| **-Y** | Dateiverschlüsselung |
| **-Yalg <encryption algorithm>** | Entspricht dem Parameter **Verschlüsselungsalgorithmus** in der GUI. Nur mit **sfs=4** (OFTP 2.0).<br>Mögliche Werte:<br>– 3DES<br>– AES |
| **-Ycsn <serial no>** | Serielle Nummer des Zertifikats für Dateiverschlüsselung |
| **-Ycin <issuer name>** | Herausgebername des Zertifikats für Dateiverschlüsselung |

Optionale remote-Parameter:

| | |
|---|---|
| **-rd <file name>** | Remote-Filename; Name der zu versendenden Datei. Die Datei muss sich im Outbox-Verzeichnis befinden. |
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

**Beispiele:**

```
createSendJob -d C:\text.txt -s RVS -v test
```

In diesem Beispiel wird die Datei `C:\text.txt` mit dem virtuellen Namen `test` an die Station `RVS` gesendet.

```
createSendJob -d C:\text.txt -s RVS -v OFTP_TEST -F F
-M 80
```

In diesem Beispiel wird die Datei `C:\test.txt` an die Station `RVS` mit dem virtuellen Namen `OFTP_TEST` gesendet; diese Datei ist eine Text-datei, in welcher jede Zeile die Länge 80 (`-F F -M 80`) ohne CR/LF hat.

```
createSendJob -d C:\part.txt -s RVS -v PART
 -S Y -l AUTO
```

In diesem Beispiel wird die Datei `C:\part.txt` an die Station `RVS` mit dem virtuellen Namen `PART` gesendet; diese Datei gehört zu der Gruppe der serialisierten Dateien mit dem Label `AUTO`.

```
createSendJob -d C:\text.txt -s RVS -v test -Sd 2010-
02-01 -St 12:30:00
```

In diesem Beispiel wird die Datei `C:\text.txt` mit dem virtuellen Namen `test` am `01.02.2010` um `12:30` h an die Station `RVS` gesendet.

```
createSendJob -d C:\text.txt -s RVS -v test -sfs 4 -C
-Y -rsr -sif
```

In diesem Beispiel wird die Datei `C:\text.txt` mit dem virtuellen Namen `test` an die Station `RVS` gesendet. Die Datei ist im Format CMS verschlüsselt, komprimiert, signiert und es wird ein signierter EERP/ NERP beantragt.

### 4.5.1 Statuswerte von Sende- und Empfangsjobs

Der folgende Abschnitt liefert einen Überblick der Statuswerte und deren Bedeutung.

**Statuswerte von Sende- und Empfangsjobs:**

– `RESTART` (Warten nach Übertragungsunterbrechung, um die Datei erneut zu senden/empfangen).
– `SP_PROCESSING` (Jobverarbeitung durch Service Provider)
– `SP_ENDED` (Jobverarbeitung durch Service Provider abgeschlossen)
– `SP_FAILED` (Verarbeitung durch Service Provider fehlgeschlagen)
– `ENDED` (Dateitransfer erfolgreich ausgeführt).
– `ENDED_WITH_JS_ERROR` (Job endete mit Fehler während des Aufrufs von Jobstart).

– FATAL_ERROR (Schwerwiegender Fehler des Empfangs-/Sendeauftrags)
– DELETED (Job wurde gelöscht)

**Statuswerte von Sendjobs:**

– CREATED (Sendeauftrag wurde erzeugt, aber noch nicht versendet)
– WF_SFID_ANSWER (Warten auf SFID Antwort für bereits gesendete SFID)
– WF_CDT (Warten auf Credit)
– WF_EFID_ANSWER (Warten auf Antwort auf bereits gesendete EFID)
– WF_EERP (Warten auf EERP)
– WF_EERP_ROUTING (Warten auf EERP bei Routing)
– SYNCHRONIZE_ERROR (Fehler bei der Synchronisation von Sendeaufträgen)
– HELD (Sendeübertragung wurde angehalten)
– RELEASED (Sendeauftrag wurde freigegeben)
– FAILED_WITH_NERP (Sendeübertragung fehlgeschlagen - Empfang eines NERP)
– FAILED_WITH_SFNA (Empfang eines SFNA mit der Einstellung Wiederaufsatz/Restart = Nein)
– FAILED_WITH_INVALID_PROTOCOL_VERSION (Übertragung fehlgeschlagen aufgrund einer ungültigen Protokollversion)
– SCHEDULED (Sendeauftrag erzeugt - Ausführungstermin noch nicht erreicht)

Statuswerte von Empfangjobs:

– RESTART_AFTER_EFNA (Datei wurde fehlerhaft übertragen; warten auf erneuten Dateiempfang vom Partner nach Versand von EFNA).
– RESTART_AFTER_EFPA_FAILURE (Datei wurde empfangen, aber EFPA war nicht erfolgreich. Der Partner wird wahrscheinlich die Datei erneut senden.)
– RECEIVING (Empfang der Daten nach SFPA)
– EERP_HELD (Die Datei wurde vollständig empfangen. EERP ist im Status HELD und kann freigegeben werden).
– EERP_RELEASED (Der Benutzer hat EERP freigegeben, aber noch nicht versendet).
– EERP_DELETED (Der Benutzer hat EERP gelöscht. Der Job wird gestoppt).
– NERP_RELEASED (Der Benutzer hat NERP freigegeben, aber noch nicht versendet)

- NERP_RELEASED_JS (NERP wegen Jobstart-Fehler freigegeben, aber noch nicht versendet)
- FAILED_WITHOUT_NERP (Übertragung fehlgeschlagen ohne Versand eines NERP)
- ROUTING_HELD_DURING_SHUTDOWN (Routing angehalten, da Server heruntergefahren)
- FAILED_WITH_INVALID_EERP_SIGNATUR (Emfang eines EERP mit ungültiger Signatur)
- FAILED_WITH_INVALID_NERP_SIGNATUR (Emfang eines NERP mit ungültiger Signatur)

**Hinweis**: Weitere Informationen erhalten Sie im Kapitel 15.1 "ODETTE-Protokoll" (z.B. über EFNA, SFID, EFID, EERP,...).

## 4.6 Synchronisation von Sendeaufträgen

Einleitung

Der Dateitransfer über OFTP wird asynchron abgearbeitet. Bei Erstellung eines Sendeauftrags wird eine Datei lediglich zum Senden bereitgestellt. Der eigentliche Versand erfolgt, sobald rvs® eine Verbindung zur Partnerstation aufbauen konnte. Gerade im automatisierten Berieb ist es jedoch oft erforderlich, direkt zu reagieren, wenn eine Datei innerhalb einer bestimmten Zeit nicht versendet werden konnte, oder der Versand erfolgreich war.

rvsEVO stellt eine Methode zur Verfügung, mit der der Dateiversand quasi synchron abgewickelt werden kann. Diese Methode kehrt erst zum Anrufer zurück, wenn der Versand erfolgreich war oder ein Fehler auftrat. Es kann die Anzahl der Sendeversuche oder die Zeit angegeben werden, innerhalb der die erfolgreiche Abwicklung des Dateiversandes erwartet wird. Wird dieser Rahmen überschritten, gilt der Versand als fehlerhaft. Der Versand gilt als erfolgreich, wenn eine OFTP-Quittung (EERP) empfangen wurde. Bei der Einstellung des OFTP-Parameters **EERP eingehend** = NEVER gilt der Versand nach Erhalt einer EFPA (**E**nd**F**ile **P**ositive **A**nswer) als erfolgreich.

Mit dieser Methode kann außerdem vor dem Dateiversand eine Konvertierung einer EDI Nachricht mit dem EDI-Konverter **WEDIConv** durchgeführt werden.

`convertAndSend`

Das Programm `convertAndSend` bietet Ihnen diese Funktionalität. Dieses Programm ist dem Programm `createSendJob` ähnlich. Es hat zusätzlich die Funktionalität zum synchronisierten Dateiversand und für die Konvertierung der EDI-Nachricht mit dem EDI-Konverter **WEDIConv**. Die Konvertierung kann in zwei Schritten erfolgen. Dem ersten Schritt kann hierbei ein XSLT-Stylesheet mitgegeben werden, mit dem das Ergebnis von dem XML-Format nochmals transformiert wird. Lesen Sie bitte das WEDIConv Benutzerhandbuch, um mehr Informationen über die Konvertierung einer EDI-Nachrichten zu erhalten.

**Syntax:**

```
convertAndSend -d <filename> -s <receiver sid>
-v <vdsn>
-[t <table name>][-F <format>]
[-M <length>][-S <serialize> -l <label>]
[-D <disposition>][-nocp][-za <attempts>]
[-zt <timeout>][converter parms] [-h?] [-verbose]
[-Sd <init date>][-St <init time>] -[Sh]
[-so <originator sid>] [-idf <filename>]
[-xid <external job id>] [-rm <record mode>]
```

Benötigte Parameter:

| | |
|---|---|
| **-d <filename>** | Dateiname der zu versendenden Datei. |
| **-s <receiver sid>** | StationsID des Empfängers. |

| | |
|---|---|
| **-v <vdsn>** | Virtueller Dateiname; das ist der Dateiname, der für die ODETTE-Übertragung benutzt wird, max. 26 Zeichen, |

Optionale Sende-Parameter:

| | |
|---|---|
| **-C** | ODETTE-Komprimierung (Offline-Komprimierung). Mögliche Werte: **Y**(Yes)/**N**(No). Standard: **N** |
| **-D <disposition>** | Diese Option legt fest, ob die versendende Datei nach erfolgreichem Versand gelöscht werden soll. Mögliche Werte: **K** (keep) - Datei wird nicht gelöscht; **D** (delete) - Datei wird gelöscht; Standard bei `ConvertAndSend`. |
| **-F <format>** | Format der zu übertragenden Datei:<br>– `T` (Text): eine Folge von ASCII-Zeichen<br>– `F` (Fest): feste Satzlänge<br>– `V` (Variabel): variable Satzlänge<br>– `U` (Unstrukturiert): binäre Datei.<br>Siehe auch Parameter **-rm** (Satzmodus). |
| **-idf <filename2>** | Die rvsEVO-JobID des erzeugten Jobs wird bei erfolgreichem Versand (return code/error level = 0) in eine vorgegebene Datei (filename2) geschrieben. |
| **-l <label>** | Alle Dateien, die in der gleichen Gruppe versendet werden, müssen die gleiche Kennung (Label) haben. Siehe auch Option `-s <serialize>`. |
| **-M <length>** | Für Dateien im Format Fest geben Sie die feste Satzlänge an, mit der der Empfänger die Datei interpretieren soll. Dies ist die Satzlänge jedes Satzes bis zum Zeilenwechsel (CR/LF bei MS Windows, LF bei UNIX-Systemen).<br>Für Dateien im Format Variabel geben Sie die maximale Satzlänge an. Siehe auch Parameter **-rm** (Satzmodus). |
| **-nocp** | Beim Erzeugen eines Sendeeintrags wird standardmäßig die zu versendende Datei ins Oubox-Verzeichnis von rvsEVO kopiert und von hier versendet. Mit der Option `-nocp` kann dies unterbunden werden. In diesem Falle wird die Originaldatei nach erfolgreichem Versand gelöscht. **Achtung!** Diese Option hat jedoch keine Auswirkung, wenn die Datei im Format `U` mit Kodekonvertierung versendet wird! |

| | |
|---|---|
| **-rm <record mode>** | In diesem Parameter können Sie für Dateien im Format 'F' oder 'V' angeben, ob es sich um eine Textdatei oder um eine Binärdatei handelt. Mögliche Werte:<br>**TXT**: Textdatei<br>**BIN**: Binäre Datei<br>Bei Textdateien im Format ’F’ werden Zeilen, die länger sind als im Parameter **Max.Satzlänge** vorgegeben, auf die definierte Länge abgeschnitten, kürzere Zeilen werden mit Leerzeichen aufgefüllt. Bei Textdateien im Format ’V’ werden längere Zeilen abgeschnitten, kürzere bleiben unverändert.<br>Bei binären Dateien haben alle Sätze außer dem letzten die maximale Satzlänge. Ist der letzte Satz kürzer als in **Max.Satzlänge** definiert, wird dieser bei Dateien im Format ’F’ mit binären Nullen aufgefüllt. |
| **-S <serialize>** | **Y**(Yes)/**N**(No) Diese Option gewährleistet, dass Ihre Dateien in einer geordneten Reihenfolge ankommen. Alle Dateien, die in der gleichen Gruppe versendet werden sollen, müssen die gleiche Kennung (Label) haben. Siehe `-l <label>`. |
| **-Sd <init date>** | Datum, an dem der Sendeauftrag ausgeführt werden soll. Eingabe im Format `<yyyy-mm-dd>` |
| **-Sh** | Sendeauftrag wird im Status `HOLD` erzeugt. (Freigabe: Siehe Kapitel 4.10.1 "Löschen, Anhalten, Freigabe von EERPs und aktiven Sendeübertragungen"). |
| **-so <originator sid>** | StationsID des Versenders: lokale (Standard) oder virtuelle Station |
| **-St <init time>** | Zeitpunkt, zu dem der Sendeauftrag ausgeführt werden soll. Eingabe im Format `<hh:mm:ss>` |

| | |
|---|---|
| **-t <table name>** | Name Ihrer Codeumwandlungstabelle.<br>Mögliche Werte:<br>Für ASCII(ANSI) - EBCDIC Umwandlung:<br>ASCII-IBM037, ASCII-IBM273, ANSI-IBM037, ANSI-IBM273.<br>Für EBCDIC - ASCII(ANSI) Umwandlung:<br>IBM037-ASCII, IBM273-ASCII, IBM037-ANSI, IBM273-ANSI<br><br>Sie haben auch die Möglichkeit, eigene Konvertierungstabellen hinzuzufügen. Sehen Sie hierzu bitte den Hinweis zu Umwandlung-stabellen auf Seite 92)) |
| **-xid <external job id>** | Parameter für die Angabe einer externen JobID. Eine externe JobID kann sich auf mehrere rvsEVO-JobIDs beziehen. |
| **-Y** | Dateiverschlüsselung; Mögliche Werte:<br>**Y**(Yes)/**N**(No).<br>Standard: **N** |
| **-za <attempts>** | Anzahl von Sendeversuchen für synchroni-sierte Übertragung. |
| **-zt <timeout>** | Timeout in Sekunden für synchronisierte Über-tragung. |

Optionale EDI-Konvertierungsparameter:

| | |
|---|---|
| **-cd** | EDI-Konverter-Richtung für den ersten Schritt, default EDI2XML. |
| **-cd2** | EDI-Konverter-Richtung für den zweiten Schritt, default XML2EDI. |
| **-ce <encoding>** | Encoding für die XML-Ausgabe; default UTF-8. |
| **-cf <Dateipfad<** | Format-Beschreibung für den ersten Schritt des EDI-Konverters. |
| **-cf2 <Dateipfad<** | Format-Beschreibung für den zweiten Schritt des EDI-Konverters (optional). |
| **-cI <0\|1>** | Einrückung für XML-Ausgabe:<br>`0` - ohne Einrückung;<br>`1` - mit Einrückung (default). |
| **-cl <Ganzzahl>** | log level für den EDI-Konverter, default:0 |

| | |
|---|---|
| **-cs <0\|1>** | Verhalten am Zeilenende, für die EDI-Nachricht-Ausgabe:<br>0 - kein LF als Segment-Trenner<br>1 - mit LF als Segment-Trenner. |
| **-ct <Dateipfad>** | Stylesheet für den ersten Schritt des EDI-Konverters. |

Optionale Parameter:

| | |
|---|---|
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-help** | Gibt Beschreibung zum aktuellen Kommando aus |
| **-?** | Fordert Hilfe an. |

**Beispiele:**

```
convertAndSend -d C:\teil56.txt -s RVS -v TEILE
-za 4
```

In diesem Beispiel wird die Datei `C:\teil56.txt` zur Station `RVS` mit dem virtuellen Dateinamen `TEILE` gesendet; die Anzahl der Sendeversuche ist begrenzt auf 4.

```
convertAndSend
-d C:\INTEGRATION\test.txt -s RVS -v TEST
-cf C:\rvsET\system\fmtDesc\fw.kanban.ineas.xml
-ct C:\rvsET\system\stylesheets\ineas2deljit.xslt
-cf2 C:\rvsET\system\fmtDesc\edifact.97.orig.xml
```

EDI-Konverter

In diesem Beispiel wird die Datei `C:\INTEGRATION\test.txt` zur Station `RVS` mit dem virtuellen Dateinamen `TEST` gesendet. Die Datei wurde auch mit dem EDI-Konverter **WEDIConv** (der in dem Verzeichnis `C:\rvsET` installiert wurde) konvertiert, zuerst vom Inhouse-Format (Datei `test.txt`) zum XML-Format (Datei `fw.kanban.ineas.xml`) und dann zur EDIFACT-Nachricht (Datei `edifact.97.orig.xml)`. Beim ersten Schritt wurde für die Darstellung folgendes Stylesheet benutzt: `ineas2deljit.xslt`.

**Beachten!**

**Hinweis**: Bei der Benutzung dieses Programms ist zu beachten, dass ein erneuter Sendeauftrag nach Fehler doppelte Übertragungen zur Folge haben kann. Wenn z.B. die Datei schon zur direkten Gegenstelle übertragen werden konnte, jedoch innerhalb der vorgegebenen Übertragungszeit die Quittung nicht eintraf, ist die Übertragung unabhängig von der eigenen Versandstation weiterhin aktiv. Es kann sein, dass die Datei von unserer direkten Partnerstation trotzdem noch übertragen wird.

Dateizähler

Darauf hat die eigene Station keinen Einfluss mehr. Ein erneutes Versenden der Datei hat dann die doppelte Übertagung zur Folge. Die Anwendung in der Schicht über rvs® muss also mit einer solchen Situation umgehen können. In der Regel wird dies gelöst, indem die Datei anhand des Dateinamens eindeutig ist, z.B. durch Verwendung

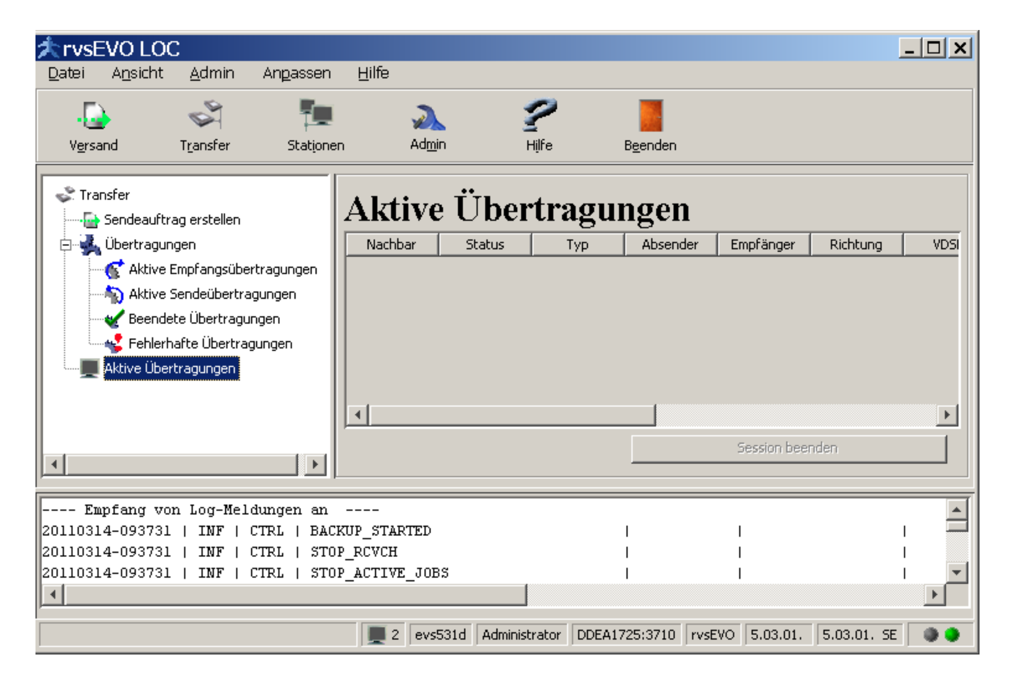
eines aufsteigenden Zählers oder indem die Anwendung die Eindeutigkeit über den Dateiinhalt feststellt.

## 4.7 Aktive Übertragungen (Active Panel)

Dieses Werkzeug vermittelt Ihnen weitere Details über den Status einer aktiven Übertragung.

In der GUI erreichen Sie den Active Panel über das Symbol Transfer:

`-> Transfer -> Aktive Sendeübertragungen`




Folgende Informationen stehen im Active Panel zur Verfügung:

Anzeige

| Parameter | Beschreibung |
|---|---|
| **Typ** | `Create Session:` Sessionaufbau<br>`Start File:` Beginn der Übertragung<br>`Transmit File:` Übertragung der Datei<br>`End File:` Ende der Übertragung |
| **Empfänger** | SID der Zielstation |
| **Richtung** | `Sending` (Senden)<br>`Receiving` (Empfang) |
| **VDSN** | virtueller Dateiname |
| **Geschw.** | Übertragungsgeschwindigkeit in kB/sec |
| **Typ** | Kommunikationstyp |

| | |
|---|---|
| **Nachbar** | Station-ID der Nachbarstation |
| **Absender** | SID der Ursprungsstation |
| **Fortschritt** | prozentualer abgeschlossener Anteil der Übertragung |
| **SessionID** | wird für jede Übertragung automatisch vergeben |
| **Start Datei** | Uhrzeit des Beginns der Übertragung der Datei. |
| **Start Session** | Uhrzeit des Verbindungaufbaus |
| **Status** | `Initiator` (eigene Station ist aktive Seite des Kommunikationsprozesses); `Responder` (eigene Station ist passive Seite des Kommunikationsprozesses) |

`Filter` Über die Schaltfläche `Filter` im unteren Bereich des Fensters ist es möglich, die Anzeige des Übertragungsstatus auf bestimmte Netzwerke zu beschränken.

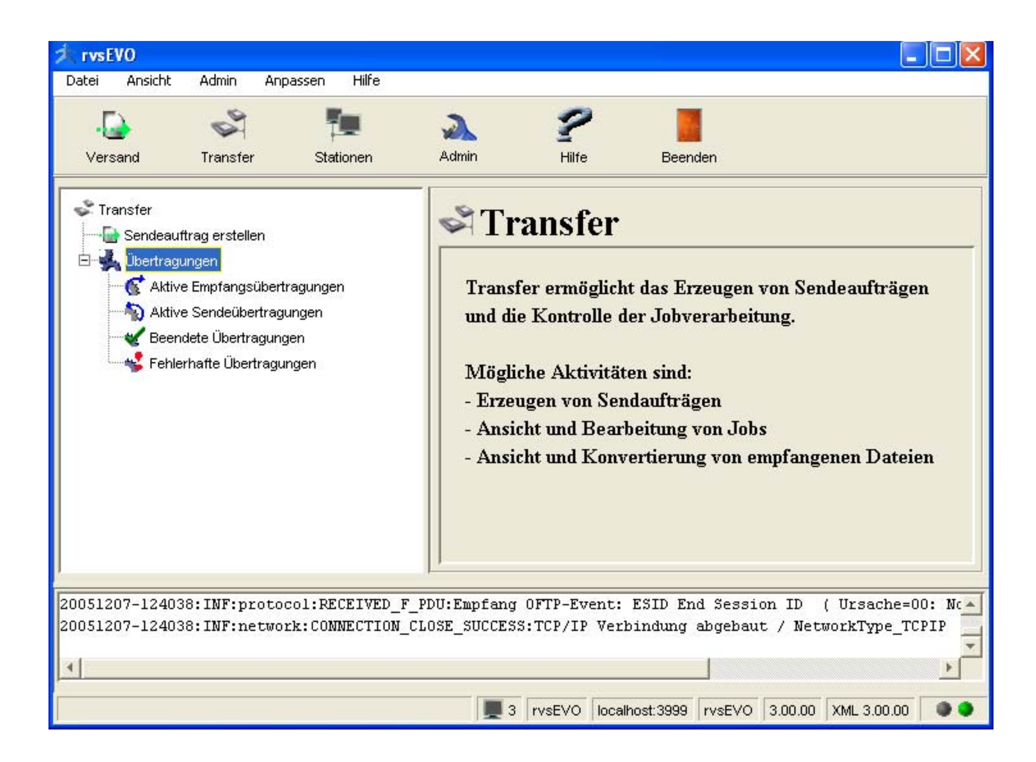
Session beenden **Hinweis**: Um eine laufende Übertragung abzubrechen, markieren Sie bitte die entsprechende Session und betätigen die Schaltfläche `Terminate Session`.

## 4.8 Ausgabe aller Empfangs- und Sendejobs

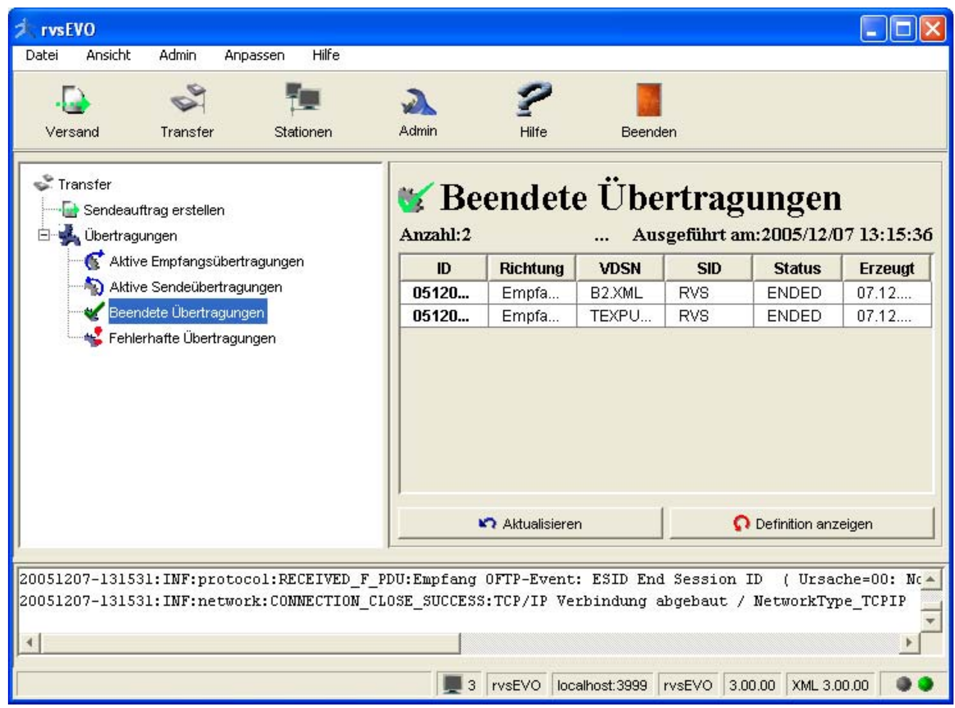
Die wichtigsten Informationen über die Sende- und Empfangsjobs können über die GUI oder mit dem Program `getJobList` angezeigt werden. Wie Sie die rvsEVO GUI starten können, lesen Sie bitte im Kapitel 2.5 "Wie starte ich rvsEVO?".

**GUI**

`Übertragungen` Wenn Sie das Symbol Transfer in der Funktionsleiste wählen, öffnet sich das Fenster mit dem Ordner `Übertragungen` in dem Transferbaum. Sie können jetzt zwischen Aktiven Empfangs/Sendeübertragungen, Beendeten Übertragungen (Sende- und Empfangs-Richtung) und Fehlerhaften Übertragungen (Sende- und Empfangs-Richtung) wählen.



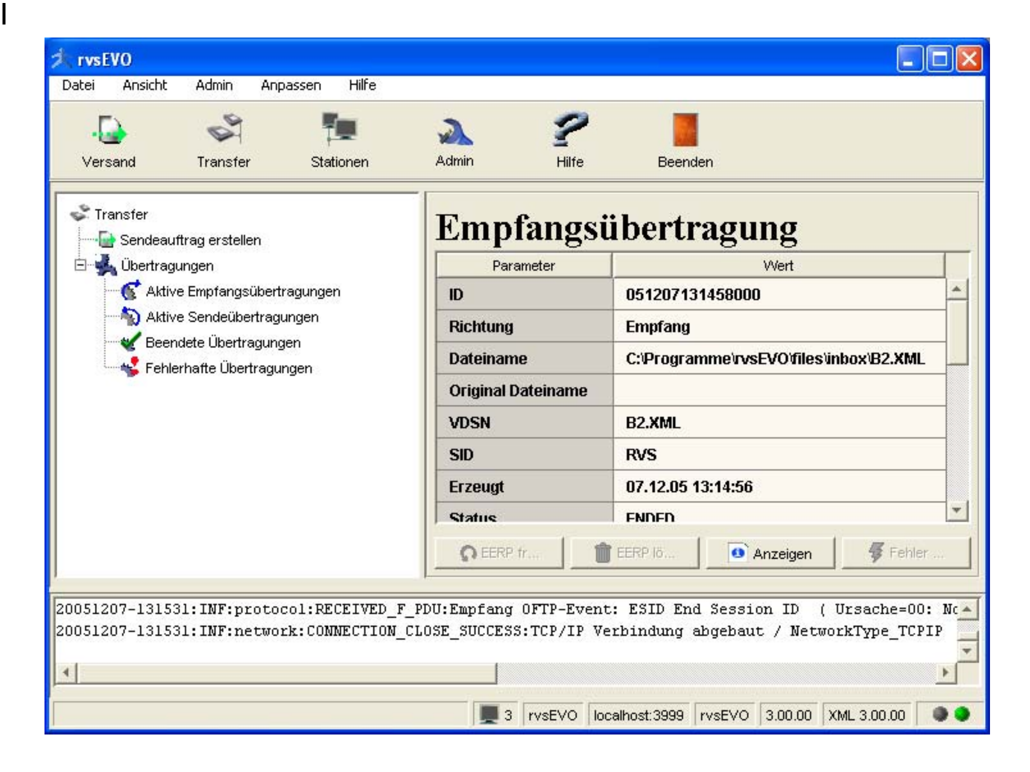

Eine Übersicht aller vorhandenen Übertragungen bekommen Sie, indem Sie den gewünschten Übertragungs-Ordner mit einem Klick anwählen (markieren).




Die Daten eines Jobs können Sie sich anzeigen lassen, indem Sie mit der Maus einen Doppelklick auf eine Jobzeile im rechten Teil des

Fensters ausführen (Detailansicht). Sehen Sie hierzu bitte auch Kapitel 4.9 "Ausgabe eines JobEintrages". Eine genaue Erläuterung der einzelnen Parameter finden Sie im Abschnitt "Erläuterungen der Jobattribute:" auf Seite 133

Beispiel




### Eingabeaufforderung

`getJobList` Mit dem Programm `getJobList` können Sie alle Jobs auflisten.

**Syntax:**

```
getJobList  [-a] [-e] [-f] [-verbose] [-help]
[-su <user> -sp <password> -sh <host:port>]
```

Optionale Parameter:

| | |
|---|---|
| **-a** | Ausführliche Ausgabe der Informationen über Jobs, die in Bearbeitung sind (zusätzlich Anzeige des Erzeugungsdatums, der SID der Nachbarstation und der VDSN). |
| **-ac S** | Zusätzliche Ausgabe des Original-Dateinamens |
| **-e** | Ausgabe der Informationen über beendete Jobs. |
| **-f** | Ausgabe der Informationen über fehlgeschlagene Jobs. |
| **-h** | Fordert Hilfe (Usage) an. |

| | |
|---|---|
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-?** | Fordert Hilfe (Usage) an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

Erfolgt der Aufruf von `getJobList` ohne Parameter, erhalten Sie eine Ausgabe der Informationen aller Jobs, die in Bearbeitung sind.

**Beispiele:**

```
getJobList -e
```

**Ergebnis:**

```
C:\Programme\rvsEVO\bin>getJobList -e
************************
job 051207131458000 (RCV): state=ENDED
job 051207131530000 (RCV): state=ENDED
************************
List ended. size=2
************************
C:\Programme\rvsEVO\bin>
```

```
getJobList -a
```

**Ergebnis:**

```
C:\rvsEVO_V2\bin>getJobList -a
**************************************************************************
Job/Transmission      Status      CreatedAt           SID:VDSN
**************************************************************************
100823143944000 (RCV) RECEIVING       23.08.10 14:39:42    EV052:TEST
100823143707000 (SND) WF_CDT       23.08.10 14:39:42    EV052:TEST2
***********************
List ended. size=2
***********************
getJobList exited with return code 0

C:\rvsEVO_V2\bin>
```

### 4.9 Ausgabe eines JobEintrages

getJob

Mit dem Programm `getJob` erhalten Sie Informationen zu einem bestimmten Sende- oder Empfangsjob.

**Syntax:**

```
getJob -n <jobid> [-a] [-verbose|-v]
[-su <user> -sp <password> -sh <host:port>]
```

Benötigte Parameter:

| | |
|---|---|
| **-n <jobid>** | Angabe eines Sende- oder Empfangsjobs mit der ID `<jobid>`. |

Optionale Parameter:

| | |
|---|---|
| **-a** | Ausgabe der komplett verfügbaren Job-Informationen. |
| **-help** | Fordert Hilfe (Usage) an. |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-?** | Fordert Hilfe (Usage) an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

**Beispiel:**

```
getJob -n 040329175603000
```

**Ergebnis**: `job 040329175603000 (SND): state: WF_EERP`

### 4.10 Joboperationen

GUI

In den Übertragungsordnern haben Sie die Möglichkeit, Operationen an Jobs durchzuführen. Sie können die Aktivitäten entweder in der Detailansicht durch Anklicken der Reiter im unteren Bereich des Fensters aktivieren oder indem Sie mit einem rechten Mausklick auf die Jobzeile ein Kontextmenü öffnen und darüber eine Auswahl treffen.

Folgende Operationen bietet rvsEVO an:

**Aktive Empfangsübertragungen:**

- EERP freigeben
- NERP freigeben
- EERP löschen
- Anzeigen
- Neustart SP
- Löschen erzwingen (Empfangsjob wird während laufender Übertragung gelöscht. Die Sendestation nimmt dies als Abbruch wahr und überträgt die Datei erneut. Diese wird mit neuer jobID aber gleichem Zeitstempel beim Empfänger gespeichert)

Bitte lesen Sie für weitere Informationen Kapitel 4.10.1 "Löschen, Anhalten, Freigabe von EERPs und aktiven Sendeübertragungen"

**Aktive Sendeübertragungen:**

- Übertragung anhalten
- Übertragung freigeben
- übertragung löschen
- Neustart SP
- Löschen erzwingen (aktiven Sendejob löschen, ohne diesen vorher anzuhalten)

Bitte lesen Sie für weitere Informationen Kapitel 4.10.1 "Löschen, Anhalten, Freigabe von EERPs und aktiven Sendeübertragungen"

**Beendete Übertragungen:**

- Anzeigen (der Datei)
- Fehler anzeigen:
- Neuerstellen: Job neu starten (nur für beendete Sendeübertragungen)

**Fehlerhafte Übertragungen:**

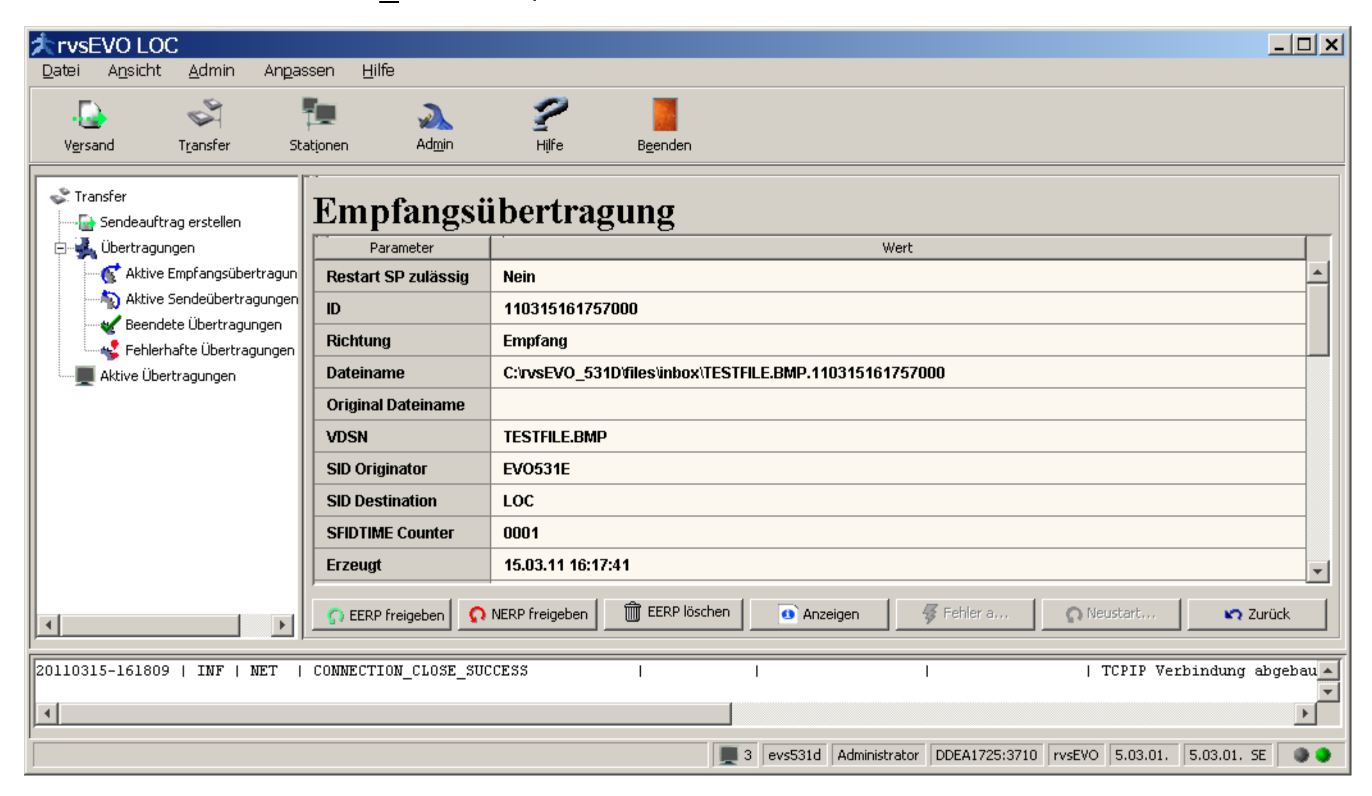
- Fehler anzeigen
  - Neuerstellen: Nachstarten einer Übertragung für die ein NERP empfangen wurde

### 4.10.1 Löschen, Anhalten, Freigabe von EERPs und aktiven Sendeübertragungen

Bei den aktiven Empfangsübertragungen bietet rvsEVO die Möglichkeiten, einen EERP freizugeben oder zu löschen. Die aktiven Sendeübertragungen können angehalten, freigegeben oder entfernt werden. Ab Version 5.02 können Sie sich zusätzlich für Jobs im Status `SP_FAILED` die Fehlermeldung anzeigen lassen und die Übertragung wieder anstoßen.

Einen EERP/NERP freigeben können Sie, indem Sie in der GUI die Maske Empfangsübertragung (Detailansicht) auswählen und anschließend im rechten Fensterteil die Schaltfläche **EERP freigeben** bzw. **NERP freigeben** betätigen und einen EERP löschen, indem Sie die

Schaltfläche **EERP löschen**, betätigen. Je nachdem für welche Aktion Sie sich entscheiden, ändert sich der Wert in der Zeile **Status** (z.B. `EERP_DELETED`).




Das Gleiche können Sie auch mit dem Programm `handleEERP` erzielen.

`handleEERP`

**Syntax:**

```
handleEERP -r|-d <JobID> [-verbose|-v]
[-su <user> -sp <password> -sh <host:port>]
```

Benötigte Parameter:

| | |
|---|---|
| **-d <JobID>** | ID des Jobs für welchen die EERP gelöscht werden soll. |
| **-n <JobID>** | ID des Jobs für welchen ein Negetive-End-to-End-Response (NERP) gesendet werden soll. |
| **-r <JobID>** | ID des Jobs für welchen die EERP freigegeben werden soll. |

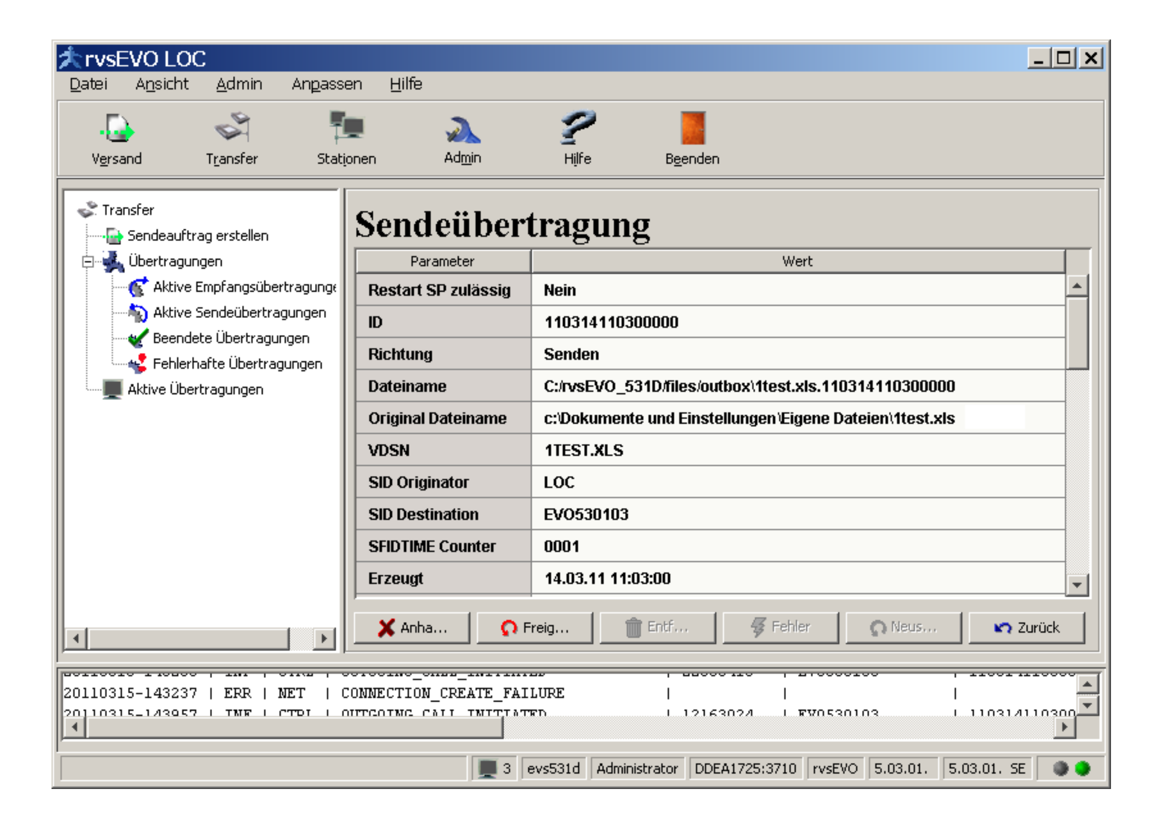
Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-verbose \| -v** | Ausgabe von ausführlichen Meldungen. |
| **-?** | Fordert Hilfe (Usage) an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

Ebenso haben Sie die Möglichkeit aktive Sendeaufträge anzuhalten, freizugeben oder zu löschen. Gehen Sie hierzu in die Detailansicht eines Sendejobs und betätigen Sie die Schaltfläche **Anhalten** um einen Sendejob anzuhalten, **Freigeben** um ihn wieder freizugeben und **Entfernen** um den Sendeauftrag zu löschen. Ein aktiver Sendejob muss zuerst angehalten werden, um gelöscht werden zu können.




Für diese Aktivitäten gibt es im Verzeichnis `$RVS_HOME/bin` folgende Kommandozeilentools:

`holdJob` einen Sendejob anhalten: `holdJob`

**Syntax:**

```
holdJob -n [-su <user> -sp <password> -sh <host:port>]
```

Benötigte Parameter:

| | |
|---|---|
| **-n** | jobID des anzuhaltenden Jobs |

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-?** | Fordert Hilfe (Usage) an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

`deleteJob` einen Job löschen: `deleteJob`

```
deleteJob -n<jobID> [-f] [-su <user> -sp <password>
-sh <host:port>]
```

Benötigte Parameter:

| | |
|---|---|
| **-n** | jobID des zu löschenden Jobs |

Optionale Parameter:

| | |
|---|---|
| **-f** | mit diesem Parameter können Sie einen aktiven Sendejob löschen, ohne diesen vorher anzuhalten. |
| **-help** | Fordert Hilfe (Usage) an. |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-?** | Fordert Hilfe (Usage) an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

`releaseJob` einen Sendejob im Status `HOLD` (angehalten) wieder freigeben: `releaseJob`.

```
releaseJob -n [-su <user> -sp <password> -sh
<host:port>]
```

Benötigte Parameter:

| | |
|---|---|
| **-n** | jobID des freizugebenden Jobs |

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-?** | Fordert Hilfe (Usage) an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

`restartJob` einen Sendejob im Status `FAILED_WITH_SFNA` oder `SP_FAILED` nachstarten: `restartJob`.

```
restartJob -n [-su <user> -sp <password> -sh
<host:port>]
```

Benötigte Parameter:

| | |
|---|---|
| **-n** | jobID des nachzustartenden Jobs |

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-?** | Fordert Hilfe (Usage) an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

**Hinweis:** Kann der Service Provider auf Grund von Zertifikatsproblemen die Daten nicht ver- oder entschlüsseln (z.B. wenn der Public Key des Partners nicht korrekt importiert wurde), wird der Status des Jobs auf `SP_FAILED` gesetzt.

Für Jobs, die sich im Status `SP_FAILED` befinden, können Sie sich durch Betätigen der Schaltfläche **Fehler** den Fehlercode und die Beschreibung anzeigen lassen. Nach der Behebung des Fehlers kann der Job wieder angestoßen und vollends abgearbeitet werden, indem Sie die Schaltfläche **Restart SP** betätigen.

### 4.11 Erzeugen einer Info-Datei zu einer externen JobID

`getJobInfoList`

Mit dem Programm `getJobInfoList` wird für eine bestimmte externe JobID eine Job-Info-Datei im XML-Format erzeugt. In diese Datei wird für jeden Job, der zu dieser externen JobID gehört, die ausführliche Jobinformation (JobExtendedDetails) geschrieben.

**Syntax:**

```
getJobInfoList -xid <external Job ID> [-f <filename>]
[-su <user> -sp <password> -sh <host:port>]
```

Benötigter Parameter:

| | |
|---|---|
| **-xid <external Job ID>** | Angabe der externen JobID, welche beim Erstellen eines Sendeauftrages vergeben werden kann. (Siehe Kapitel 4.5 "Versand einer Datei") |

Optionaler Parameter:

| | |
|---|---|
| **-f <filename>** | Vorgabe eines Namens für Info-Datei.<br>Ohne Angabe des Parameters, wird die Datei `joblist.xml` im aktuellen Verzeichnis erzeugt. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

Das XML-Schema `JobInfoList.xsd` befindet sich im Verzeichnis `$RVS_HOME/system/data/xsd`.

Job-Info-List-Datei

Das root-Element der Datei ist `JobInfoList`. Alle Information über einen Job sind im XML-Element `JobExtendedDetails` zu finden. Es können mehrere `JobExtendedDetails` in einer Datei vorhanden sein (alle Jobs, die zu einer externen JobID gehören). Ein XML-Element `JobExtendedDetails` besteht aus `JobDetails` und `JobError`.

Das XML-Element `JobError` ist optional. Es besteht aus `ErrorIdentifier`, `ErrorText` und `ErrorType`.

**Erläuterungen der Jobattribute:**

| Job-Attribut | Beschreibung |
|---|---|
| **Code-Tabelle <conversionTable>** | Pfad der Konvertierungstabelle für Konvertierung von ASCII nach EBCDIC und umgekehrt. |
| **Dateibeschr. <fileDescription>** | Datei-Beschreibung. Textkommentar, möglich ab OFTP Version 2.0. Wenn Ihre Gegenstelle kein OFTP 2.0 kann, wird dieses Feld ignoriert. |
| **Dateiname <filename>** | Der ganze Pfad der Datei, die aus dem rvsEVO-outbox-Verzeichnis versendet wird. Beim Versand wird die Datei aus dem Originalverzeichnis ins outbox-Verzeichnis kopiert und dann versendet. |
| **Dateisignatur <sign>** | Signatur; mögliche Werte: true (Ja) oder false (Nein). Standard: false |
| **<deleteAllowed>** | Dieser Parameter gibt Auskunft darüber, ob das Löschen eines Jobs abhängig von seinem Status erlaubt ist. Mögliche Werte: true (Ja) oder false (Nein). Standard: false |
| **Encryption Cert Issuer Name <encryptionCertificateIssuerName>** | Herausgebername des Zertifikats. Ein Wert wird nur angezeigt, wenn ein Zertifikat während der Erzeugung eines Sendeauftrages explizit für die Verwendung angegeben wurde. Es wird nicht ausgegeben, welches Zertifikat tatsächlich verwendet wird. |
| **Encryption Cert Serial No. <encryptionCertificateSerialNumber>** | Serielle Nummer des Zertifikats. Ein Wert wird nur angezeigt, wenn ein Zertifikat während der Erzeugung eines Sendeauftrages explizit für die Verwendung angegeben wurde. Es wird nicht ausgegeben, welches Zertifikat tatsächlich verwendet wird. |
| **<errorID>** | Wenn der errorType Transmission ist, sind diese Fehler-Codes identisch mit denen von ESID, EFNA, SFNA des Odette-Protokolls. |
| **<errorText>** | Wenn der errorType Transmission ist, sind diese Fehler-Texte identisch mit denen von ESID, EFNA, SFNA des Odette-Protokolls. |
| **<errorType>** | Typ des Fehlers; Mögliche Werte: Internal, Transmission, Missing Receipt, Undefined |

| | |
|---|---|
| **Erzeugt**<br>**<creationDate>** | Datum und Uhrzeit der Erstellung eines Sendeauftrages (aus SFID) in der Form TT/MM/JJ HH:MM:SS (in Verindung mit **SFIDTIME Counter**) |
| **Externe JobID**<br>**<externalJobId>** | JobID für externe Applikationen |
| **<filePos>** | Anzahl der gelesenen Bytes der Sendedatei |
| **Format**<br>**<recordFormat>** | Format der zu übertragenden Datei:<br>T (Text): eine Folge von ASCII-Zeichen,<br>F (fest): feste Satzlänge,<br>V (variabel): variable Satzlänge,<br>U (unstrukturiert): binäre Datei. |
| **<holdAllowed>** | Dieser Parameter gibt Auskunft darüber, ob das Anhalten eines Jobs, abhängig von seinem Status, erlaubt ist. Mögliche Werte: true (Ja) oder false (Nein). Standard: false |
| **ID**<br>**<jobNumber>** | rvsEVO-JobID |
| **Komprimierung**<br>**<compression>** | Komprimierung; mögliche Werte: true (Ja) oder false (Nein). Standard: false |
| **Label**<br>**<label>** | Lable für Serialisierung. Alle Dateien, die in der gleichen Gruppe versendet werden, müssen die gleiche Kennung (Label) haben. Wenn dieser Parameter bei Serialisierung=Y (true) nicht gesetzt ist, wird VDSN als Label verwendet. |
| **Länge der Originaldatei**<br>**<lengthOriginFile>** | Länge der Originaldatei beim Senden in Bytes vor der Verarbeitung durch ServiceProvider. Beim Empfang wird dieser Parameter auf 0 gesetzt. |
| **<lastByteRead>** | Wert des letzten gelesenen Bytes einer zu versendenden Datei in Dezimaldarstellung als Zeichenkette. Standard: -1 |
| **<lastByteSend>** | Wert des letzten gesendeten Bytes einer zu versendenden Datei in Dezimaldarstellung als Zeichenkette. Standard: -1 |
| **Letzte Änderung**<br>**<lastStateChange>** | Zeitpunkt der letzten Änderung des Jobstatus (zu dem die Datei dem lokalen Benutzer zugestellt wurde) im Format TT.MM.JJ HH:MM:SS |
| **Nerp Creator OdetteID**<br>**<nerpCreatorOdetteID>** | OdetteID der Station, die einen NERP sendet |
| **Nerp Reason Code**<br>**<nerpReasonCode>** | Code des NERP |

| | |
|---|---|
| **Nerp Reason Text <nerpReasonText>** | Beschreibung des NERP |
| **<oidNeighbor>** | OdetteID der Nachbarstation |
| **Original Dateiname <filenameSrc>** | Der ganze Pfad der Originaldatei; diese Datei wird in das rvsEVO-Outbox-Vereichnis kopiert. |
| **<outputCharset>** | |
| **<recCount>** | Anzahl der gesendeten Datensätze (records) |
| **<releaseAllowed>** | Dieser Parameter gibt Auskunft daüber, ob das Freigeben eines Jobs abhängig von seinem Status erlaubt ist. Mögliche Werte: true (Ja) oder false (Nein). Standard: false |
| **Restart Position <restartPos>** | Wiederaufsatzpunkt (restart position), bei dem eine erneute Übertragung der Datei starten soll. |
| **Richtung <direction>** | Richtung; mögliche Werte `0` (Senden) oder `1` (Empfangen). |
| **Satzlänge <recordLength<** | Max. Satz-Länge |
| **<scheduleDateTime>** | geplante Zeit für den Jobstart in der Fom TT.MM.JJ HH:MM:SS |
| **Sendeversuche <sendAttempts>** | Anzahl der fehlgeschlagenen Sendeversuche. Steht hier „0", so bedeutet dies erfolgreichen Dateiversand. |
| **Serialisierung <serialisation>** | Diese Option bedeutet, dass Ihre Dateien in einer geordneten Reihenfolge ankommen. Alle Dateien die in der gleichen Gruppe versendet werden sollen, müssen die gleiche Kennung (Lable) haben. Mögliche Werte: true (Ja) oder false (Nein). Standard: false |
| **SFIDTIME Counter** | fortlaufender Zähler, wenn mehrere Sendeaufträge zeitgleich erstellt werden (in Verbindung mit **Erzeugt**) |
| **Sicherheitsmerkmale <securityFeatureSet>** | Sicherheitsmerkmale; mögliche Werte: 1 (keine), 2 (ComSecure V1), 3 (ComSecure V2) oder 4 (OFTP 2.0) |
| **<oidDest>** | OdetteID der Zielstation |
| **<oidOrig>** | OdetteID der Sendestation (Originator) |
| **SID Destination <sidDestination>** | Station-ID der Zielstation |

| | |
|---|---|
| **SID Originator** **<sidOriginator>** | Station-ID der Ursprungsstation |
| **<SID>** | Station-ID abhängig von der Richtung beim Senden: SID des Empfängers beim Empfangen: SID des Versenders |
| **Sign Cert Issuer Name** **<signCertificateIssuerName>** | Herausgebername der Signatur. Ein Wert wird nur angezeigt, wenn ein Zertifikat während der Erzeugung eines Sendeauftrages explizit für die Verwendung angegeben wurde. Es wird nicht ausgegeben, welches Zertifikat tatsächlich verwendet wird. |
| **Sign Cert Serial No.** **<signCertificateSerialNumber>** | Serielle Nummer der Signatur. Ein Wert wird nur angezeigt, wenn ein Zertifikat während der Erzeugung eines Sendeauftrages explizit für die Verwendung angegeben wurde. Es wird nicht ausgegeben, welches Zertifikat tatsächlich verwendet wird. |
| **signierten EERP/NERP beantragen** **<signEERP>** | signierten EERP beantragen; Mögliche Werte: true (Ja) oder false (Nein). Standard: false |
| **Status** **<status>** | Jobstatus. Für mögliche Statuswerte siehe Kapitel 4.5 "Versand einer Datei" |
| **Time Start File** **<timeStartFile>** | Zeit des Transferanfangs (Systemzeit in Millisekunden) |
| **<transferFileLength>** | Senden: Größe der zu sendenden Datei. Empfang: Geschätze Größe (SFID) der zu empfangenden Datei in Bytes |
| **Übertragene Bytes** **<transmittedBytes>** | Anzahl der tatsächlich übertragenen Bytes |
| **<user>** | für zukünftige Anwendungen |
| **VDSN** **<VDSN>** | virtueller Dateiname für Odetteübertragung |
| **Verbleib** **<disposition>** | Verbleib; Dieser Prameter entscheidet, ob die zu versendende Datei nach dem Versand bei Ihnen lokal gelöscht wird oder erhalten bleibt. Mögliche Werte: K (keep): Datei bleibt erhalten oder D (delete): Datei wird gelöscht. Standard: K |
| **Verschlüsselung** **<encryption>** | Verschlüsselung; mögliche Werte: true (Ja) oder false (Nein). Standard: false |
| **Verschlüsselungsalgorithmus** **<encryptionAlgorithm>** | Verschlüsselungsalgorithmus; mögliche Werte: leer (kein Algorithmus), DES_DED3_CBC (3DES) oder AES256_CBC. |

| | |
|---|---|
| **<virtualSID>** | für zukünftige Anwendungen |
| **<waitTime> (TransmissionFailWaitTime)** | Zeit in Millisekunden für einen erneuten Start nach einem Misserfolg Muss gesetzt werden im Falle von SFNA (Start File Negative Answer) oder von EFNA (End File Negative Answer). |

**Beispiel:**


```
<JobInfoList>
  <JobExtendedDetails>
    <JobDetails>
      <externalJobId>5006</externalJobId>
      <JobID>
        <direction>0</direction>
        <jobNumber>091222132218000</jobNumber>
      </JobID>
      <compression>false</compression>
      <conversionTable></conversionTable>
      <creationDate>22.12.09 13:22:18</creationDate>
      <deleteAllowed>false</deleteAllowed>
      <disposition>K</disposition>
      <direction>SND</direction>
      <encryption>false</encryption>
      <encryptionCertificateSerialNumber></encryptionCertificateSerialNumber>
      <encryptionCertificateIssuerName></encryptionCertificateIssuerName>
      <fileDescription></fileDescription>
      <filename>C:/rvsEVO/files/outbox\rvsenv.dat.091222132218000</filename>
      <filenameSrc>C:\rvs\rvsenv.dat</filenameSrc>
      <filePos>789</filePos>
      <holdAllowed>false</holdAllowed>
      <lastStateChange>22.12.09 13:22:20</lastStateChange>
      <lastByteRead>-1</lastByteRead>
      <lastByteSend>-1</lastByteSend>
      <lengthOriginFile>789</lengthOriginFile>
      <oidDest>OXP</oidDest>
      <oidNeighbor>OXP</oidNeighbor>
      <oidOrig>OGEDASEVO</oidOrig>
      <recordFormat>U</recordFormat>
      <recordLength>0</recordLength>
      <recCount>0</recCount>
      <releaseAllowed>false</releaseAllowed>
      <restartPos>0</restartPos>
      <scheduleDateTime></scheduleDateTime>
```

```
        <sendAttemps>0</sendAttemps>
        <serialisation>false</serialisation>
        <SID>XP</SID>
        <sidOriginator>LOC</sidOriginator>
        <sidDestination>XP</sidDestination>
        <signCertificateSerialNumber></signCertificateSerialNumber>
        <signCertificateIssuerName></signCertificateIssuerName>
        <status>ENDED</status>
        <timeStartFile>1261484539412</timeStartFile>
        <transmittedBytes>789</transmittedBytes>
        <transferFileLength>789</transferFileLength>
        <user></user>
        <VDSN>WEIH6644</VDSN>
        <virtualSID></virtualSID>
        <waitTime>0</waitTime>
        <jobSecurityAndSign>
          <securityFeatureSet>1</securityFeatureSet>
          <encryptionAlgorithm></encryptionAlgorithm>
          <sign>false</sign>
          <signEERP>false</signEERP>
        </jobSecurityAndSign>
      </JobDetails>
    </JobExtendedDetails>
  </JobInfoList>
```


## 4.12 Archivieren der Einträge der abgearbeiteten Sende- bzw. Empfangsaufträge im Revisionlog

Bei der Archivierung werden die Einträge der abgeschlossenen und fehlgeschlagenen Sende- und Empfangsaufträge aus der rvsEVO-Datenbank gelöscht und in die Datei `RevisionLog.xml` im Verzeichnis `$RVS_HOME\archive` geschrieben. Der Name der RevisionLog-Datei setzt sich zusammen aus „RevisionLog.xml.“, Zeitstempel und fortlaufendem Zähler beginnend mit „0000“ (Format: RevisionLog.xml.yyMMddHHmmss_cccc). Mittels der rvsEVO-Umgebungsparameter **MaxRevisionLogSize** und **MaxRevisionLogCount** können Sie die maximale Größe und maximale Anzahl der RevisionLog-Datein festlegen. Bei erreichen der im Parameter **MaxRevisionLogSize** angegebenen Größe wird eine neue RevisionLog-Datei generiert. Standardmäßig findet alle 24 Stunden eine Archivierung der Jobs statt, die älter sind als 7 Tage. Diese Einstellung ist mittels der Parameter **CleanupDays**, **CleanupInterval** und **CleanupTime** konfigurierbar (Siehe “rvsEVO-Umgebungsparameter” auf Seite 41.).

Alternativ haben Sie die Möglichkeit, über den Menüpunkt `Admin -> Archivierung` einen Archivierungsvorgang auszulösen und dabei das Alter der zu archivierenden Sende- und Empfangsaufträge anzugeben.

`archiveJobs` Über die Eingabeaufforderung (Kommandozeile) können Sie die Archivierung mit dem Programm `archiveJobs` starten.

**Hinweis**: Wenn Sie die Archivierung bei laufendem rvsEVO ausführen möchten, verwenden Sie die Option **-r**.

**Syntax:**

```
archiveJobs -d <date>[-r][-help] [-?]
[-verbose|-v]
```

Benötigter Parameter:

| | |
|---|---|
| **-d <date>** | Dieser Parameter definiert ein Datum in der Form `yyMMddHHmmss` oder `yyMMdd` mit dem festgelegt wird, dass nur Übertragungen archiviert werden, die älter als angegebenes Datum sind. |

Optionale Parameter:

| | |
|---|---|
| **-r** | die Archivierung wird bei laufendem rvsEVO ausgeführt. |
| **-help** | Fordert Hilfe (Usage) an |
| **-?** | Fordert Hilfe (Usage) an. |
| **-verbose (-v)** | Ausgabe von ausführlichen Meldungen. |

## 4.13 Zertifikat ausliefern

Mit diesem Tool kann ein bestimmtes Zertifikat für die Dateiverschlüsselung (aus `$RVS_HOME\system\data\keystore.p12`) an eine Partnerstation ausgeliefert und bei der Partnerstation importiert werden. Der virtuelle Dateiname (VDSN) ist `ODETTE_CERTIFICATE_DELIVER`.

**Syntax:**

```
deliverCertificate -s <stationID>
[-i <keyIndex>]-r <stationID>
[-su <user> -sp <password> -sh <host:port>][-help]
[-?] [-verbose|-v]
```

Benötigte Parameter:

| | |
|---|---|
| **-r <stationID>** | StationID der Partner-Station, an die das Zertifikat gesendet werden soll. |
| **-s <stationID>** | StationID der Station, deren Zertifikat gesendet werden soll. |

Optionale Parameter:

| | |
|---|---|
| **-i <keyIndex>** | Index des Zertifikats, welches gesendet werden soll, wenn mehrere Zertifikate für eine Station vorhanden sind. Wenn dieser Parameter nicht gesetzt ist, werden alle vorhandenen Zertifikate gesendet. |
| **-?** | Fordert Hilfe (Usage) an. |
| **-verbose (-v)** | Fordert Hilfe (Usage) an. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

### 4.14 Zertifikat anfordern

Mit diesem Tool kann ein Zertifikat für die Dateiverschlüsselung (aus `$RVS_HOME\system\data\keystore.p12`) von einer Partnerstation angefordert werden. Der virtuelle Dateiname (VDSN) ist `ODETTE_CERTIFICATE_REQUEST`.

**Syntax:**

```
requestCertificate -s <stationID>
[-su <user> -sp <password> -sh <host:port>]
[-help] [-?][-verbose|-v]
```

Benötigter Parameter:

| | |
|---|---|
| **-s <stationID>** | StationID der Partner-Station, von der Zertifikate angefordert werden. |

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |
| **-verbose (-v)** | Ausgabe von ausführlichen Meldungen. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-Login |
| **-sp <password>** | dessen Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

### 4.15 Zertifikat ersetzen

Mit diesem Tool kann ein Zertifikat für die Dateiverschlüsselung (aus `$RVS_HOME\system\data\keystore.p12`) an eine Partnerstation ausgeliefert werden und ein bei dieser Station schon vorhandenes Zertifikat ersetzen. Der virtuelle Dateiname (VDSN) ist `ODETTE_CERTIFICATE_REPLACE`.

**Syntax:**

```
replaceCertificate -s <stationID>
[-i <keyIndex>]-r <stationID>
[-su <user> -sp <password> -sh <host:port>]
[-help] [-?][-verbose|-v]
```

Benötigte Parameter:

| | |
|---|---|
| **-r <stationID>** | StationID der Partner-Station, an die das eigene Zertifikat (Zertifikate) gesendet wird. Das alte (zu der eigenen Station beim Partner zugehörige) Zertifikat (Zertifikate) wird ersetzt und für ungültig erklärt. |
| **-s <stationID>** | StationID der eigenen Station, deren Zertifikat gesendet und ersetzt werden soll. |

Optionale Parameter:

| | |
|---|---|
| **-i <keyIndex>** | Index des Zertifikats, welches gesendet und ersetzt werden soll. Wenn dieser Parameter nicht gesetzt ist, werden alle vorhandenen Zertifikate gesendet. Anschließend werden beim Partner alle alten Zertifikate durch neue ersetzt. |
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |
| **-verbose (-v)** | Ausgabe von ausführlichen Meldungen. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-Login |
| **-sp <password>** | dessen Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

### 4.16 Liste der Zertifikate anzeigen

Mit diesem Tool kann eine Liste der Zertifikate (Dateiverschlüsselung) einer Station angezeigt werden.

Syntax:

```
getCertificateList -sid <stationID> [-help] [-?]
```

Benötigte Parameter:

| | |
|---|---|
| **-sid <stationID>** | StationID der Station, deren Zertifikate angezeigt werden sollen. |

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |

### 4.17 Schlüsselverwaltungsdatei öffnen

Mit dem Kommandozeilentool `startKeyMgn.bat` kann die Schlüssel-verwaltungsdatei geöffnet werden.

Syntax:

```
startKeyMgn [-h] [-?]
```

Optionale Parameter:

| | |
|---|---|
| **-h** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |

### 4.18 Eine CRL importieren

Mit diesem Tool kann eine CRL (Certificate Revocation List) für eine bestimme Partnerstation importiert werden.

Voraussetzungen: Um dieses Tool ausführen zu können, muss das Zertifikat der Partner-Station von einer CA (Certification Authority) ausgestellt worden sein. Beim Aufruf von `importCRL` wird von der im Zertifikat der Partner-Station hinterlegten Internetadresse die aktuelle CRL heruntergeladen. Zuvor müssen Sie das Verzeichnis, in dem die CRL gespeichert werden soll, im Parameter **location** in der PKI-Konfigurationsdatei `$RVS_HOME/conf/PkiParameter.xml` bestimmen. (Für weitere Informationen siehe Kapitel 11.2.2 "PKI-Konfigurationsdatei")

**Syntax:**

```
importCRL -s <SID> [-i <keyIndex>]
[-help] [-?][-verbose|-v]
```

Benötigter Parameter:

| | |
|---|---|
| **-s <stationID>** | StationID der Partner-Station, deren CRL importiert werden soll. |

Optionale Parameter:

| | |
|---|---|
| **-i <keyIndex>** | Index des Zertifikats, wenn mehrere Zertifikate für eine Station vorhanden sind. Wenn dieser Parameter nicht gesetzt ist, wird CRL für alle vorhandenen Zertifikate der Partnerstation importiert. |
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |
| **-verbose (-v)** | Ausgabe von ausführlichen Meldungen. |

**Hinweis:** Die Proxy-Einstellungen sind in der Datei `$RVS_HOME/conf/rvs-system.properties` mittels folgender Definitionen konfigurierbar:

– `http.proxyHost:` IP-Adresse oder Hostname des Proxy-Servers

- `http.proxyPort`: Port, über den die Verbindung zum Proxy-Server aufgebaut werden soll
- `http.nonProxyHosts`: Hosts von denen ein CRL-Download ohne Proxy-Server erlaubt ist
- `http.proxyUser`: System-Benutzername
- `http.proxyPassword`: Passwort des Benutzers

### 4.19 Eine TSL importieren

Mit diesem Tool kann eine TSL (Trust Service Status List) in die Schlüsselverwaltungsdatei (keystore) importiert werden.

**Syntax:**

```
importTSL -f <TSL filename> -k <keystore filename>
-u <TSL URL>[-help] [-?][-verbose|-v]
```

Benötigte Parameter:

| | |
|---|---|
| **-f <TSL filename>** | kompletter Pfad der Datei, die TSL enthält. |
| **-k <keystore filename>** | kompletter Pfad der Schlüsselverwaltungsdatei |

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |
| **-verbose (-v)** | Ausgabe von ausführlichen Meldungen. |

### 4.20 Eine Session beenden

Mit diesem Tool kann eine aktive Session beendet werden.

**Syntax:**

```
terminateSession -s <session_id>
[-su <user> -sp <password> -sh <host:port>]
[-help] [-?][-verbose|-v]
```

Benötigter Parameter:

| | |
|---|---|
| **-s <session_id>** | ID der Session, die beendet werden soll. |

**Hinweis**: Eine SessionID finden Sie im MonitorLog (`$RVS_HOME/log/monitor.log` oder im ActivePanel.)

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |
| **-verbose (-v)** | Ausgabe von ausführlichen Meldungen. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-Login |
| **-sp <password>** | sein Passwort |
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

### 4.21 Erzeugen und Senden eines Journals

Mit dem Tool `sendJournal.bat` können Sie ein Journal erzeugen und dieses an das Zentrale Journal senden.

**Syntax:**

```
sendJournal -d <date> -f <file name> [-su <user> -sp
<password>] [-sh <host:port>] [-verbose] [-help]  [-?]
```

Benötigte Parameter:

| | |
|---|---|
| **-d <date>** | Datum im Format yyMMddHHmm |
| **-f <filename>** | Name der Request-Datei vom Zentralen Journal. Default: ZJREQUEST |

Optionale Parameter:

| | |
|---|---|
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |
| **-verbose (-v)** | Ausgabe von ausführlichen Meldungen. |

Optionale remote-Parameter:

| | |
|---|---|
| **-su <user>** | Client-Benutzer für remote-Login |
| **-sp <password>** | sein Passwort |

| | |
|---|---|
| **-sh <host:port>** | Verbindungsparameter für remote-Login (**host** ist der Rechnername oder die IP-Adresse des rvsEVO-Servers und **port** ist der Port des rvsEVO-Servers. |

Weitere Informationen finden Sie im Benutzerhandbuch „Zentrales Jounal".

### 4.22 Konvertieren von Datein

Mit dem Programm `convertFile` können Sie Dateiformate von ASCII(ANSI) zu EBCDIC und andersherum überführen.

Erforderliche Parameter:

| | |
|---|---|
| **-c <Konvertierungsbe-zug>** | Die möglichen Werte sind im Parameter **Conversion Table** in der Tabelle "Unterelemente von Jobfilter" auf Seite 86 beschrieben. |
| **-s <Dateiname>** | Pfad und Name der Ursprungsdatei, die konvertiert werden soll |
| **-d <Dateiname>** | Pfad und Name der Zieldatei |

Optionaler Parameter:

| | |
|---|---|
| **-?** | Fordert Hilfe (Usage) an. |

### 4.23 Kommandozeilentools zur internen Verwendung

Die folgenden Programme dienen ausschließlich der internen Verwendung und bedürfen daher keiner weiteren Beschreibung:

```
login.bat
rvsEVOService.exe
setclientcp.bat
setcp.bat
userManagerClient.bat
```

# 5 Sicherung und Wiederherstellung der rvsEVO-Daten

rvsEVO bietet die Möglichkeit, eine Sicherung aller relevanten Daten durchzuführen und bei Bedarf diese Sicherung wieder einzuspielen. Dieses Verfahren ist inbesondere dann von Bedeutung, wenn ein Fehler in rvsEVO aufgetreten ist und der Benutzer den alten Zustand vor dem Fehler wiederherstellen möchte.

## 5.1 Sicherung (Backup)

Eine Sicherung kann nur durchgeführt werden, wenn der rvsEVO-Server gestartet ist.

Das Sicherungsprogramm beendet alle aktiven Übertragungen vor dem Backup. Nach dem Backup werden die abgebrochenen Übertragungen wieder angestoßen. Es ist kein Neustart des Servers erforderlich.

Die Funktion Sicherung kann entweder in der rvsEVO GUI mit dem Menüpunkt `Admin -> Backup` durchgeführt werden oder durch Aufruf des Skriptes `createBackup` auf der Kommandozeile gestartet werden.

**Syntax:**

```
createBackup [-d <dir>] [-verbose][-help][-?]
```

Alle Parameter sind optional.

| | |
|---|---|
| **-d <dir>** | Angabe des Sicherungsverzeichisses; ohne diese Option werden die Sicherungsdaten ins Verzeichnis `$RVS_HOME/archive` geschrieben. |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |

**Hinweis**: Mit dem Parameter `BackupOnStartup` kann konfiguriert werden, dass bei jedem Start von rvsEVO eine automatische Sicherung durchgeführt wird. Dieser Parameter ist über die grafische Oberfläche (Admin -> Parameter -> Umgebung) oder in der Konfigurationsdatei `$RVS_HOME/conf/rvsConfig.xml` zu setzen. Standardwert ist `Y` (Yes).

### 5.1.1 Was wird gesichert?

Folgende Dateien oder Verzeichnisse werden gesichert:

– Das ganze Verzeichnis `$RVS_HOME/conf`, welches die Konfigurationsdateien von rvsEVO beinhaltet.

- Sicherung der Daten aus der rvsEVO-Datenbank (Jobdaten). Im Unterverzeichnis `Job` entstehen Sicherungdateien. Standardmäßig wird der Datenbank-Backup in die Datei `dbbackup.xml` geschrieben. Die Jobdatentabelle heißt `HIB_JOB_DATA`. Die Sicherungsdatei kann wieder mit der Funktion **Recover** eingespielt werden. Gleichzeitig entstehen im Verzeichnis auch eine dtd-Datei (`dbbackup.dtd`) und die Dateien (mit der Endung `.bin`) für die binary Parameter aus der Jobdaten-Tabelle. Solche Parameter sind z.B. `SIGNATURE` und `FILE DESCRIPTION`. Auf diese Dateien wird in der Sicherungsdatei verwiesen.
  **Beispiel**: `HIB_JOB_DATA_FILEDESCRIPTION_1.bin`
- Dateien aus dem Verzeichnis `$RVS_HOME/system/data`.

Aus den gesicherten Daten wird eine `<BackupZeitpunkt>.jar`-Datei erstellt. Mit `BackupZeitpunkt` ist der Dateiname gemeint, der automatisch bei der Sicherung vergeben wird und der Form YYMMDDHHMMSS mit einem zusätzlichen 3-stelligen Zähler besitzt.

**Beispiel**: Datei `051010112417000.jar` ist die Sicherungsdatei, die am 10.10.05 um 11:24:17 h erstellt wurde. Es wurden bisher keine anderen Dateien zu diesem Zeitpunkt erstellt und daher lautet der zusätzliche Zähler `000`.

### 5.1.2 Redo-Log

Ab dem Zeitpunkt der Sicherung werden alle dynamischen Daten in einem fortlaufenden Log (Redo-Log) protokolliert.

Mit „dynamischen Daten“ sind die Informationen über die Sende- und Empfangsjobs gemeint. Diese Daten werden geschrieben, um die nicht vollendeten Übertragungsjobs später wiederherstellen zu können. Die vollendeten Jobs werden protokolliert, sind aber nicht von Bedeutung für die Wiederherstellung (Siehe 5.2).

Damit das Redo-Log eindeutig einer Sicherung zugeordnet werden kann, erhält es den gleichen Namen (Zeitstempel) wie die zugehörige Sicherungsdatei allerdings mit dem Präfix `Redo_` und der Erweiterung `.log`. Diese Datei wird genauso wie die Sicherungsdatei, in das Verzeichnis `$RVS_HOME/archive` geschrieben.

**Beispiel**:

Zu der Sicherungsdatei `051010112417000.jar` wurde die zugehörige Redo-Logdatei `Redo_051010112417000.log` erzeugt.

Sobald eine neue Sicherung erzeugt wurde, wird auch eine zugehörige Redo-Logdatei angelegt. Dabei wird bei allen vorhandenen Redo-Logdateien die Dateierweiterung `.log` durch `.old` ersetzt.

Die Redo-Logdatei ist eine Textdatei. Jede Zeile enthält ein XML-Element `<Job>` mit allen zugehörigen Jobinformationen wie z.B. `ID`, `FileName`, `VDSN`, `SID`.

## 5.2 Wiederherstellung der rvsEVO-Daten (Recovery)

Eine Wiederherstellung der rvsEVO-Daten erfolgt über die grafische Oberfläche, Menüpunkt `Admin/Recover`. In dem Dialog, welcher dann angeboten wird, kann man den Namen der Sicherungsdatei (Sicherung der rvsEVO-Datenbankdaten) und des Redo-Logs angeben.

Wenn die Angabe des Dateinamens ohne Pfad erfolgt, werden die Sicherungsdatei und das Redo-Log im Verzeichnis `$RVS_HOME/archive` gesucht.

Alternativ kann diese Funktion auf der Kommandozeile mit dem Skript `$RVS_HOME/bin/doRecover.bat` gestartet werden.

**Syntax:**

```
doRecover -b <dir> -r <name>
[-help][-?]
```

| | |
|---|---|
| **-b <dir>** | Pflichtparameter: Name der Sicherungsdatei; kann als kompletter Dateiname inklusive Verzeichnis oder als Dateiname ohne Verzeichnis angegeben werden. Im letzteren Fall wird die Sicherungsdatei im Verzeichnis `$RVS_HOME/archive` erwartet. |
| **-r** | Name der Redo-Logdatei. Er kann als kompletter Dateiname inklusive Verzeichnis oder als Dateiname ohne Verzeichnis angegeben werden. Im letzteren Fall wird die Redo-Log-Datei im Verzeichnis `$RVS_HOME/archive` erwartet. |
| **-verbose** | Ausgabe von ausführlichen Meldungen. |
| **-help** | Fordert Hilfe (Usage) an. |
| **-?** | Fordert Hilfe (Usage) an. |

**Hinweis**: Vor dem Ausführen des Wiederherstellungsprozesses müssen sowohl alle Sende- und Empfangsprozesse als auch der Service Provider (Verschlüsselung/Komprimierung) beendet werden.

Das Einspielen des Redo-Logs wird in der Datei `$RVS_HOME/log/monlog.log` protokolliert.

Nach Ausführung der Recover-Funktion ist ein Neustart von rvsEVO notwendig.

# 6 Verschlüsselte Übertragung mit rvsEVO (Dateiverschlüsselung)

In diesem Kapitel werden die Grundlagen der Dateiverschlüsselung mit zwei Verschlüsselungsformaten (Com-Secure und CMS) und die Verwaltung der Schlüssel für den sicheren Dateiaustausch in rvsEVO beschrieben.

In rvsEVO ab der OFTP Version 2 wird für die Verschlüsselung auf der Session-Ebene (Leitungsverschlüsselung) das Protokoll TLS (Transport Layer Security) eingesetzt. Lesen Sie bitte das Kapitel 3.2 "Anpassen der Stationskonfiguration", um mehr darüber zu erfahren, wie Sie rvsEVO für TLS konfigurieren sollen. Das Prinzip der Schlüsselverwaltung (öffentliche und privater Schlüssel) ist das Gleiche für Com-Secure, CMS und für TLS.

## 6.1 Einleitung: Grundlagen

rvsEVO bietet zwei Formate für die Dateiverschlüsselung an: Com-Secure und CMS.

Com-Secure ist ein eigenes Format, welches auch für die Verschlüsselung in rvs® portable verwendet wird.

CMS (Cryptographic Message Syntax) ist ein Internet-Standard (rfc 2630), welcher den Aufbau einer verschlüsselten Nachricht beschreibt. Dieser Standard wird erst ab OFTP Version 2 unterstützt.

In Com-Secure und CMS werden die Vorzüge des symmetrischen und des unsymmetrischen Verfahrens kombiniert: die hohe Geschwindigkeit des symmetrischen Verfahrens und das Sicherheitsniveau des unsymmetrischen Verfahrens. rvsEVO setzt folgende Verfahren ein:

- **3DES** als symmetrisches Verfahren für ComSecure (Länge 3x56 Bit = 168 Bit)
- **3DES** und **AES** als symmetrische Verfahren für CMS
- **RSA** als unsymmetrisches Verfahren für ComSecure und CMS (Länge 768 bis 2048 Bit).

Elektronische Signatur

Zur Erhöhung der Sicherheit arbeitet die Verschlüsselungskomponente mit elektronischer Signatur. Mit der Signatur wird sichergestellt, dass die Daten bei der Übertragung keine unbemerkte Veränderung erfahren.

Ab der OFTP Version 2 erstreckt sich die elektronische Signatur auch auf den End-To-End-Response (EERP) und den Negative-End-to-End-Response (NERP).

**Hinweis**: Beachten Sie, dass ohne JCE (Java Cryptography Extension) kein verschlüsselter Dateiaustausch möglich ist. Folgende Fehlermeldung erscheint, wenn Sie versuchen, die Dateien verschlüsselt auszutauschen, ohne die JCE-Dateien installiert zu haben: „Invalid key length“. Weitere Informationen zu diesem Thema, finden Sie im Kapitel 2.3 "Neuinstallation von rvsEVO" und in der Datei `$RVS_HOME/docu/liesmich.txt`.

## 6.2 Prinzip und Ablauf der rvsEVO-Verschlüsselung

Die Prinzipien der Verschlüsselung, die im nächsten Abschnitt erläutert werden, sind grundsätzlich die gleichen für die beiden Verschlüsselungsformate (CMS und ComSecure).

Jeder Teilnehmer der verschlüsselten Kommunikation erstellt auf seinem System ein Schlüsselpaar, bestehend aus dem **öffentlichen Schlüssel** und dem **privaten Schlüssel**.

Öffentlichen Schlüssel verteilen / privaten Schlüssel sicher aufbewahren

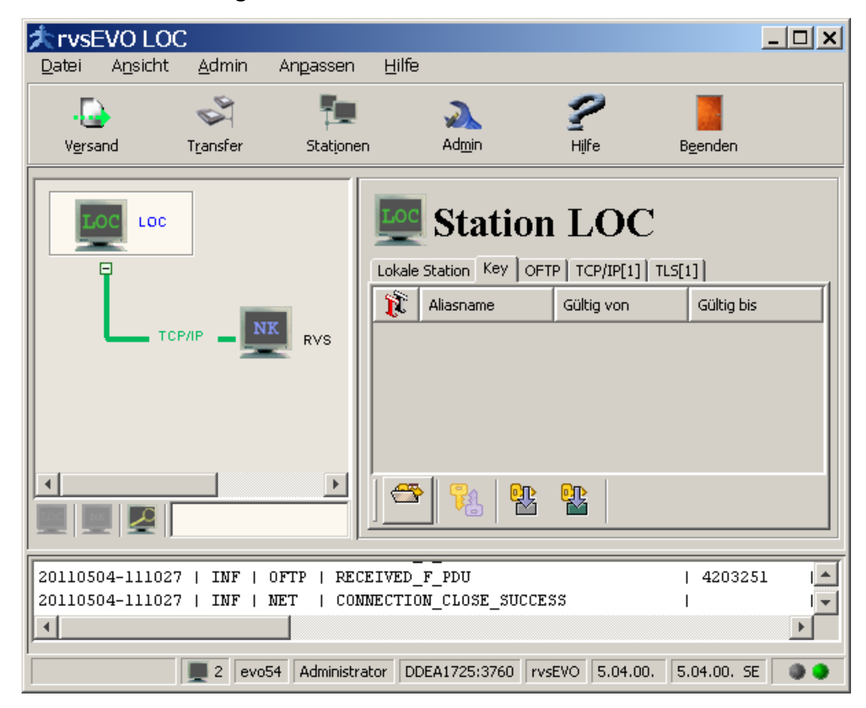
Den öffentlichen Schlüssel stellt er jedem Partner zur Verfügung, von dem er Dateien erwartet. Der jeweilige Sender kann damit die Daten genau für den Partner verschlüsseln, von dem dieser öffentliche Schlüssel stammt.

Den privaten Schlüssel des Schlüsselpaars behält jeder Teilnehmer für sich und bewahrt ihn sicher auf. Dieser wird für die Entschlüsselung der vom Partner verschlüsselten Dateien benötigt. Desweiteren wird der private Schlüssel auch zur Erzeugung einer Signatur verwendet.

In den nächsten Abschnitten wird die Schlüsselverwaltung für die Dateiverschlüsselung beschrieben.

Zur Schlüsselverwaltung der Dateiverschlüsselung gelangen Sie, indem Sie die Registerkarte **Key** in der Stationsverwaltung der GUI öffnen.

Es öffnet sich folgendes Fenster:




Folgende Funktionalitäten stehen zur Verfügung:

- Liste der Schlüssel und Zertifikate anzeigen
- Schlüsselpaar generieren
- Zertifikat importieren

– Öffentlichen Schlüssel für ComSecure importieren (nicht über Remote-GUI)
– Zertifikatsdetails anzeigen
– Zertifikat exportieren (nicht über Remote-GUI)
– Zertifikat zu ComSecure exportieren (nicht über Remote-GUI)
– Schlüssel löschen
– Zertifikatsanfrage erzeugen

Folgende Schritte sind durchzuführen, um verschlüsselte Dateien mit rvsEVO versenden zu können:

- Generieren Sie Ihr eigenes Schlüsselpaar (Siehe Kapitel 6.3)
- Exportieren Sie den eigenen öffentlichen Schlüssel in Form von X.509 Zertifikaten (Siehe Kapitel 6.4)
- Senden Sie Ihren eigenen öffentlichen Schlüssel zu Ihrem Partner (per E-Mail oder per rvsEVO)
- Ihr Partner muss Ihren öffentlichen Schlüssel (public key) in sein OFTP-Produkt importieren und Ihnen sein X.509 Zertifikat zusenden.
- Wenn sie das X.509-Zertifikat (den öffentlichen Schlüssel) vom Partner erhalten haben, importieren Sie es in die Schlüsselverwaltung von rvsEVO (Siehe 6.4).

Wie Sie eine Datei versenden und dabei Dateiverschlüsselung als Option einschalten können, lesen Sie bitte im Kapitel 4.5 "Versand einer Datei".

### 6.3 Wie erzeuge ich ein eigenes Schlüsselpaar?

Gehen Sie über die Maske Ihrer lokalen Station in die Schlüsselverwaltung.

Im unteren Teil des Schlüsselverwaltungsfensters finden Sie die Schaltflächen

- Schlüssel und Zertifikate anzeigen
- Schlüsselpaar erzeugen
- Zertifikat importieren
- ComSecure öffentlichen Schlüssel importieren

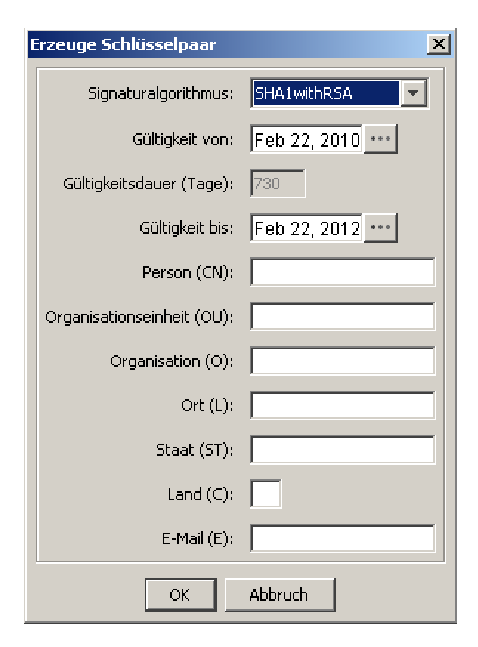
Wählen Sie den Button zum Erzeugen eines neuen Schlüsselpaars:



Im Fenster **Erzeuge Schlüsselpaar** soll als Schlüssel-Algorithmus (`Key Algorithm`) `RSA` und als Schlüsselgröße (`Key Size`) die Standardeinstellung 1024 gewählt werden.




Nach Bestätigung mit **OK** öffnet sich das Dialogfenster **Erzeuge Schlüsselpaar**:



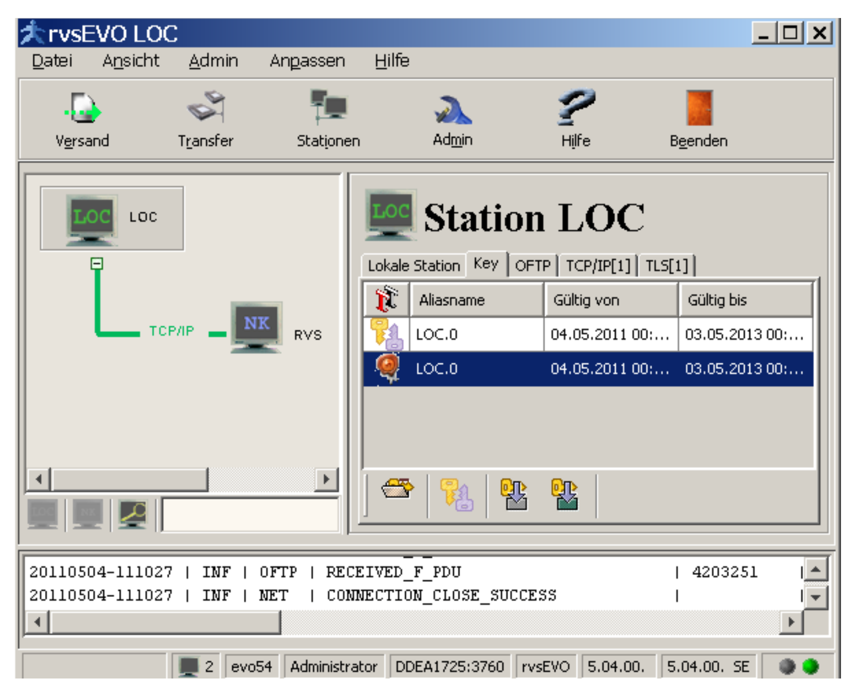

Als Signatur-Algorithmus (`Signature Algorithm`) soll `SHA1withRSA` gewählt werden. Die Angabe des Gültigkeitszeitraumes ist zwingend erforderlich.

Der Parameter `Person (Common Name)` ist z.B. von Bedeutung, wenn es sich um eine Anbindung an eine vorhandene PKI (Public Key Infrastructure) handelt und diese Angabe für LDAP-Struktur benötigt wird. Dieser Parameter ist obligatorisch.

Die restlichen Angaben betreffen Ihre Organisation. rvsEVO macht hier keine Vorschriften, wie diese Felder auszufüllen sind.

Nachdem Sie alle Angaben im Fenster **Erzeuge Schlüsselpaar** ausgefüllt und mit **OK** bestätigt haben, wird das Schlüsselpaar erzeugt und im rechten Fenster erscheint je eine Zeile mit der Angabe der Gültigkeitsdauer des privaten und öffentlichen Schlüssels.

**Hinweis:** Es besteht die Möglichkeit, die ODETTE-ID und den DNS-Namen in das Schlüsselpaar einzubauen. Diese Funktionalität ist in der Datei `$RVS_HOME\conf\rvs-system.properties` mittels der Definitionen `rvs_evo.certificate.odette-id.include=LOC` (SID für welche die ODETTE-ID eingefügt werden soll) und `rvs_evo.certificate.dns-name.include=oftp2.t-systems.com` (DNS-Name) konfigurierbar. Im exportierten Zertifikat sind diese Werte unter Details im Feld `Alternativer Antragstellername` ablesbar.




Die Verwaltung der eigenen (privaten und öffentlichen) und der (öffentlichen) Partner-Schlüssel erfolgt in einer Schlüsselverwaltungsdatei (keystore).

Der Pfad der zu verwendenden Schlüsselverwaltungsdatei ist über die GUI oder in der Konfigurationsdatei `$RVS_HOME/conf/cryptoParameter.xml` (XML-Element: `keyStoreParameter`; Unterelement: `fileName`) zu setzen. Standardwert ist: `$RVS_HOME/system/data/keystore.p12`.

**Hinweis**: Die Endung `.p12` bedeutet, dass diese Schlüsselverwaltungsdatei im Format PKCS #12 ist.

`CertificateUsageDefinition`

Wenn Sie mehrere eigene Schlüsselpaare erzeugt haben, können Sie in der xml-Datei `$RVS_HOME/conf/CertificateUsageDefinition` bestimmen, welcher private Schlüssel für welche Aktionen verwendet werden soll. Folgende Aktivitäten können gewählt werden:

– `F-ERP-SIGN` => EERP/NERP-Signatur (OFTP2)
– `F-SIGN` => Datei-Signatur (OFTP2)
– `F-COMSECURE-SIGN` => Datei-Signatur (ComSecure)

**1. Beispiel:**

Die lokale Station `LOCALSTATION` soll den Schlüssel mit der seriellen Nummer `4D623B3B` zur Datei-Signatur (OFTP2) verwenden:

```
<entry key="LOCALSTATION.F-SIGN.SERIALNUMBER">4D623B3B</
entry>
```

**2. Beispiel:**

Die virtuelle Station `VIRTUALSTATION` soll den Schlüssel mit der seriellen Nummer `4D623B3B` zur EERP/NERP-Signatur verwenden:

```
<entry key="VIRTUALTATION.F-ERP-SIGN.SERIALN-
UMBER">4D623B3B</entry>
```

### 6.4 Zertifikat importieren und exportieren

rvsEVO bietet die Möglichkeit vertrauenswürdige Zertifikate von Partnern in die Schlüsselverwaltung zu importieren und schon importierte Zertifikate aus der Schlüsselverwaltung in eine Datei zu exportieren.

Um ein X.509-Zertifikat (öffentlicher Schlüssel) in die rvsEVO-Schlüsselverwaltung zu importieren, klicken Sie auf die Schaltfläche **Zertifikat importieren**.



Das Fenster **Vertrauenswürdiges Zertifikat importieren** öffnet sich und Sie können die Datei auswählen, die importiert werden soll. Nachdem Sie **OK** gedrückt haben, ist das Zertifikat im Fenster Schlüsselverwaltung sichtbar.

Umgekehrt ist es auch möglich, den eigenen öffentlichen Schlüssel oder ein schon importiertes Zertifikat als eine Zertifikatsdatei zu exportieren. Markieren Sie für diese Zwecke das Zertifikat, welches exportiert werden soll



und betätigen Sie die rechte Maustaste. Wählen Sie aus dem Kontextmenü den Eintrag **Zertifikat exportieren**. Das Fenster **Zertifikat exportieren** öffnet sich und Sie können den Ordner auswählen, in welchen das Zertifikat exportiert werden soll.

### 6.5 ComSecure-Schlüssel importieren und exportieren

Da rvsEVO auch ein eigenes Format für öffentliche Schlüssel unterstützt, ist es notwendig, beim Import/Export die Konvertierung aus/in ein X.509-Zertifikat  (Standard-Format für digitale Zertifikate; Datenamenserweiterung `.cer`) vorzunehmen.

Es gibt zwei Funktionen, die diese Funktionalität unterstützen. Diese sind: **Öffentlichen Schlüssel für ComSecure importieren** und **Zertifikat zu ComSecure exportieren**.

Mit der ersten Funktion **Öffentlichen Schlüssel für ComSecure importieren** ist es möglich, einen öffentlichen Schlüssel im ComSecure-Format als ein X.509-Zertifikat in die rvsEVO-Schlüsselverwaltung zu importieren. Diese Funktion ist erreichbar über die Schaltfläche am unteren Ende des Schlüsselverwaltungsfensters:



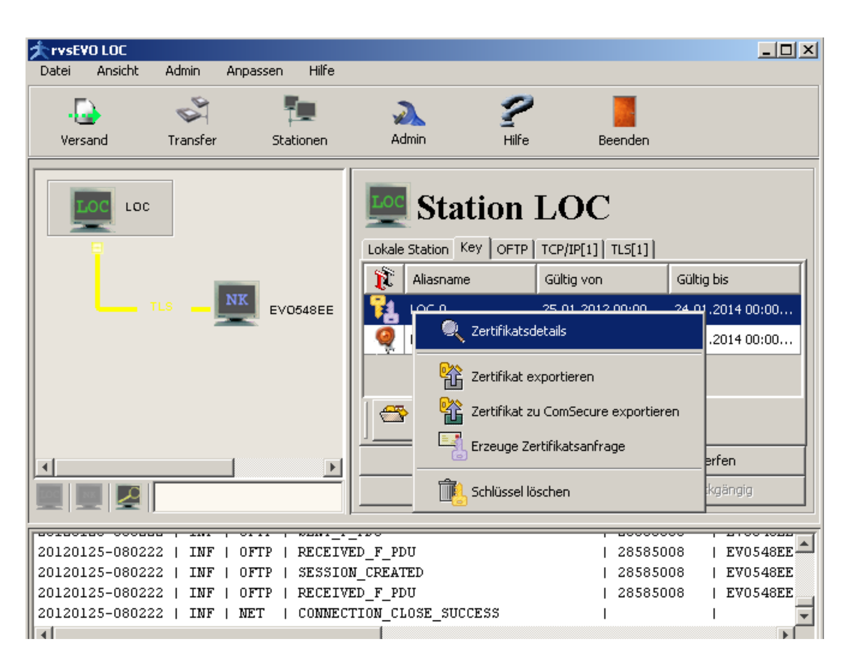
Die Funktion **Zertifikat zu ComSecure exportieren** ermöglicht, ein in rvsEVO schon vorhandenes Zertifikat als öffentlichen Schlüssel im ComSecure-Format zu exportieren. Diese Funktion ist erreichbar, indem Sie mit der rechten Maustaste auf das markierte Zertifikat klicken, und im Kontextmenue **Zertifikat zu ComSecure exportieren** auswählen.

### 6.6 Dateiverschlüsselung unter Verwendung von CA-Zertifikaten

CA-Zertifikate (**C**ertificate **A**uthority) werden von einem zertifizierten Dienstleister (TrustCenter), z. B. der Odette Zertifizierungsstelle, ausgestellt. Bitte gehen Sie wie folgt vor, wenn Sie mit CA-Zertifikaten arbeiten:

CSR generieren

- Generieren Sie ein Schlüsselpaar, wie in Kapitel 6.3 beschrieben.
- Erzeugen Sie anschließend eine Zertifikatsregistrierungsanforderung (CSR - **C**ertificate **S**igning **R**equest), indem Sie mit der rechten Maustaste auf Ihr Schlüsselpaar klicken und die Option `Erzeuge Zertifikatsanfrage` wählen. Name und Verzeichnis, in dem der Request abgelegt werden soll, sind frei wählbar. Die Datei wird standardmäßig im Format PKCS#10 erzeugt.



Beispiel eines Zertifizierungsrequests:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBWDCBwgIBADAbMRkwFwYDVQQDDBBldm81NDhkLW9yaS1maWxlMIGfMA0GCSqGS
A4GNADCBiQKBgQCczTte+HyTxy/lCCjtXWuADq07H/LDDKfH7GZANKS2WqJ9+Ozw7
Tcik5JS3Re8L3q6xGBzDxucD7N6Qp3ikincYTxaE92yuy+rbK43YJJRLdgixKAu+a
YY/a2UjKOeGY+oVRmWJLlu+pbIs9VAStMwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBA
p426CHVOREM6QzVWf2qNe40R/mxhuKT118kP1xx9oG4K7pkPbHiMuT1SdNmzb/GDp
tisBm185o+fCDOKaOpIKidjC9aTact16YtSoRdc4v7RIw5tlzpRw+ANhwOwVwZMRg
/pK79MbM
-----END NEW CERTIFICATE REQUEST-----
```

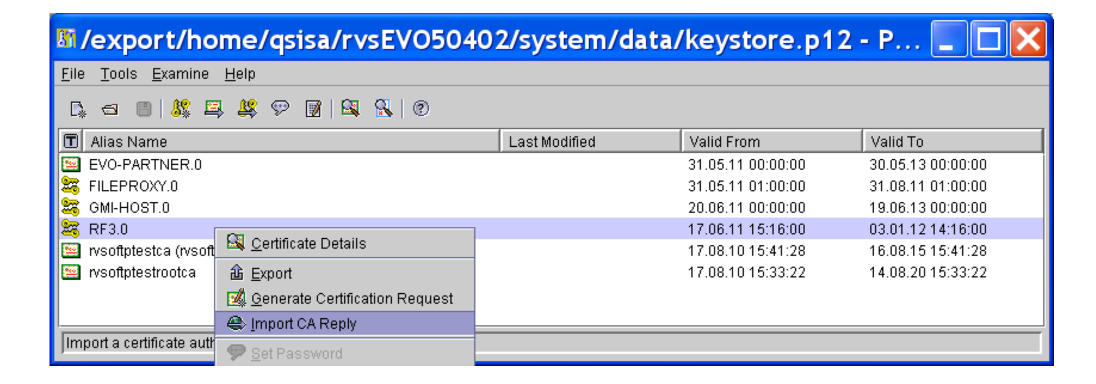
- Senden Sie den Zertifikatsrequest an Ihren Zertifizierungsdienstleister. Bei der Odette-Zertifizierungsstelle können Sie den Antrag Online stellen: `https://www.odetteca.com`. Ausführliche Informationen finden Sie im Internet: `https://forum.odette.org/repository/odette-ca-help.pdf`.

Der Antrag wird von der Zertifizierungsstelle überprüft, welche bei einem positiven Ergebnis ein signierters Serverzertifikat erstellt. Anschließend erhalten Sie von der CA neben Ihrem Serverzertifikat zwei weitere Zertifikate: ein CA-Root- und ein CA-Zertifikat.

eigene Zertifikate importieren

Importieren Sie zuerst das CA-Root- und anschließend das CA-Zertifikat mit dem Tool `startKeyMgn.bat` (`startKeyMgn.sh` bei Unix-Systemen) in die Datei `$RVS_HOME/system/data/keystore.p12`.

- Starten Sie das Programm `startKeyMgn` (siehe Kapitel 4.17 "Schlüsselverwaltungsdatei öffnen"). Das Tool portecle öffnet die Schlüsselverwaltungsdatei `$RVS_HOME/system/data/keystore.p12`.
- Importieren Sie zuerst das CA-Root-Zertifikat mit der Funktion **Tools -> Import Trusted Certificate**
- Importieren Sie dann das CA- Zertifikat mit der Funktion **Tools -> Import Trusted Certificate**
- Importieren Sie zum Schluss das Serverzertifikat mit der Funktion **Import CA Reply** (importiere CA Antwort), indem Sie das Schlüsselpaar markieren, für welches die CA-Reply importiert werden muss, und nach Betätigung der rechten Maustaste **Import CA Reply** auswählen.



In diesem Beispiel wurden das CA-Root- und das CA-Zertifikat schon in die Datei `keystore.p12` importiert. An der Liste der Schlüssel und Zertifikate ändert sich auf der Oberfläche nichts. Aber in der Detailanzeige des Schlüsselpaares, für welches die CA-Reply importiert wurde, werden die drei Zertifikate angezeigt (Certificate 1 of 3).

(Beispiel: Schlüsselpaar für RF3 -> rechte Maustaste -> Certificate Details)

- Speichern Sie den Keystore und beenden Sie diesen anschließend.

Senden Sie Ihre Zertifikate auch Ihrem Partner zu und importieren Sie ebenfalls dessen Zertifikate.

Zertifikat des Partners importieren

Die Vorgehensweise ist abhängig von der Einstellung des OFTP-Parameters **Zertifikatsvalidierung**. Wenn **Zertifikatsvalidierung** = `CERT_PATH` ist, gehen Sie bitte wie folgt vor.

- Starten Sie das Programm `startKeyMgn` (siehe Kapitel 4.17 "Schlüsselverwaltungsdatei öffnen").
- Importieren Sie das zuerst das CA-Root-Zertifikat, dann das CA-Zertifikat und am Ende das Serverzertifikat mit der Funktion **Tools -> Import Trusted Certificate** in die Datei `$RVS_HOME/system/data/keystore.p12`.
- Sichern Sie den Keystore und beenden diesen anschließend.

Bei einer anderen Einstellung des Parameters **Zertifikatsvalidierung** müssen Sie nur das Serverzertifikat importieren. Gehen Sie in diesem Fall bitte vor, wie in Kapitel **6.4 "Zertifikat importieren und exportieren"** beschrieben.

**Hinweis:** Für weitere Informationen zur Zeritfikatsvalidierung sehen Sie bitte Kapitel 11 "PKI-Anbindung mit rvsEVO"

## 6.7 Empfang ohne vorherige Entschlüsselung

Mit ComSecure verschlüsselte Dateien können wahlweise ent- oder verschlüsselt und komprimiert oder dekomprimiert empfangen werden. Diese Funktionalität wird in der properties-Datei `$RVS_HOME/conf/rvs-system.properties` durch folgende Definition gesteuert:

```
rvs_evo.serviceprovider.receive_job.process_comsecure=false
```

`false` = ausschalten, d.h. keine Entschlüsselung / Dekomprimierung

`true` = Standard-Verhalten (Entschlüsselung / Dekomprimierung)

Nicht entschlüsselte oder dekomprimierte Dateien werden nach Empfang in dem Verzeichnis `$RVS_HOME/files/inbox` gespeichert.

Die spätere Entschlüsselung / Dekomprimierung wird mit dem Programm ComSecure vorgenommen. (Sehen Sie bitte das Benutzerhandbuch ComSecure)

## 6.8 Schlüssel und Zertifikate löschen

Diese Funktion erreichen Sie, indem Sie das Schlüsselpaar oder Zertifikat, welches gelöscht werden soll, markieren und mit der rechten Maustaste im Kontextmenü die Funktion **Schlüssel löschen** aufrufen.

# 7 rvs® OFTP Proxy

In diesem Kapitel wird die Konfiguration und die Funktionalität der rvs® OFTP Proxy-Anbindung in rvsEVO erläutert.

## 7.1 Grundlagen

Internet

Das Internet wird zunehmend genutzt, um auch kommerzielle und unternehmenskritische Daten beliebigen Inhalts zu übertragen. Dafür sind solche Vorteile ausschlaggebend, wie seine große Verbreitung, die enormen verfügbaren Bandbreiten und die günstigen Übertragungskosten.

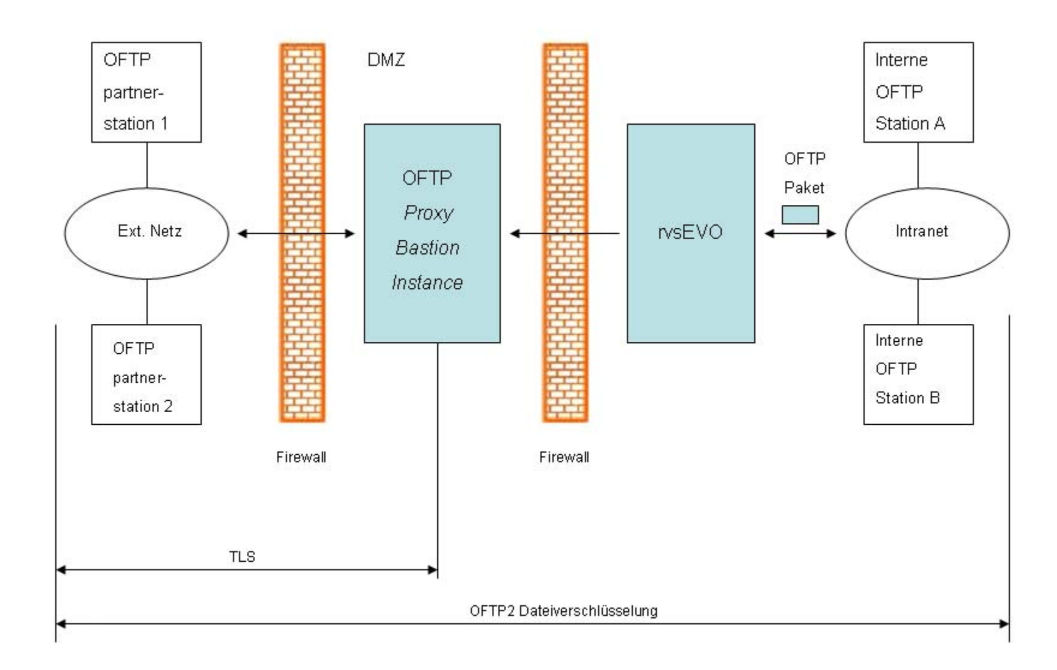
Diesen Vorzügen stehen ernst zu nehmende Gefahren gegenüber: im Internet sind alle Daten Angriffen auf ihre Vertraulichkeit und Integrität ausgesetzt. Ihre Authentizität ist ebenfalls nicht ohne weiteres sichergestellt. Das kann dazu führen, dass Daten von ihrem augenscheinlichen Absender verleugnet werden (repudiation). Solchen Gefahren wird mit geeigneten Protokollen, wie TLS und HTTPS begegnet. Für CAD Daten, wie auch EDI -und beliebige andere Daten ist die OFTP Version 2 als sicheres Internet-Protokoll standardisiert worden.

Ein weiteres Risiko besteht durch Malware, die im Internet in großem Umfang und automatisiert verbreitet wird. Viren, Würmer, Trojaner und andere schädliche Programme können beträchtliche Schäden in der IT-Infrastruktur von Unternehmen anrichten. Um dieses Risiko zu minimieren, werden Netzbereiche geschaffen, die von den Unternehmensnetzen durch Firewalls getrennt sind (auch demilitarized Zone - DMZ genannt). In den DMZ werden beispielsweise http-Proxys eingesetzt, die Nutzdaten transportieren, ohne direkte Verbindungen zwischen dem Internet und dem Intranet zuzulassen. Eine ähnliche Lösung bietet rvs® OFTP Proxy, der in der DMZ eingesetzt werden kann, um kontrolliert Daten des OFTP Protokolls Version 1 und 2 zwischen Intranet und Internet zu übertragen.

## 7.2 rvs® OFTP Proxy Architektur

Zwei Instanzen

Der rvs® OFTP Proxy wird mit zwei Instanzen betrieben: der Bastion Instance und eine in rvsEVO integrierte Connection Instance. Die folgende Abbildung soll diesen Sachverhalt veranschaulichen:




Über die **Bastion Instance** können Verbindungen aus dem externen Netz entgegengenommen werden. Diese externen Verbindungen werden an rvsEVO weitergeleitet.

Andersherum kann rvsEVO über die Bastion Instance Verbindung zu OFTP-Stationen im externen Netzwerk aufbauen.

Zwischen rvsEVO und der Bastion Instance wird ausschließlich über von rvsEVO aufgebaute Verbindungen kommuniziert. Die Kommunikation beschränkt sich auf einen Port, den konfigurierbaren RMI-Port der Bastion Instance.

Proxy-Listener

Damit rvsEVO für die externen Stationen erreichbar ist, startet rvsEVO einen Proxy-Listener. Über diesen Proxy-Listener wird eine Verbindung zu der zu erreichenden Partnerstation hergestellt.

Alle fachlichen Konfigurationen sowie TLS-Schlüssel und Zertifikate werden ausschließlich in rvsEVO hinterlegt.

Die Bastion Instance wird mit dem Programm rvs® OFTP Proxy installiert. Sie können rvs® OFTP Proxy über folgende Webseite herunterladen: https://servicenet.t-systems.de/tsi/de/267072/Startseite/Business-Integration/rvs

Falls Ihnen das nicht möglich ist, wenden Sie sich bitte an Ihren Vertriebspartner:

Telefon aus dem Inland: 0800 664 77 45;
Telefon international: +375 606 19 902
E-Mail: rvs-service@t-systems.com.

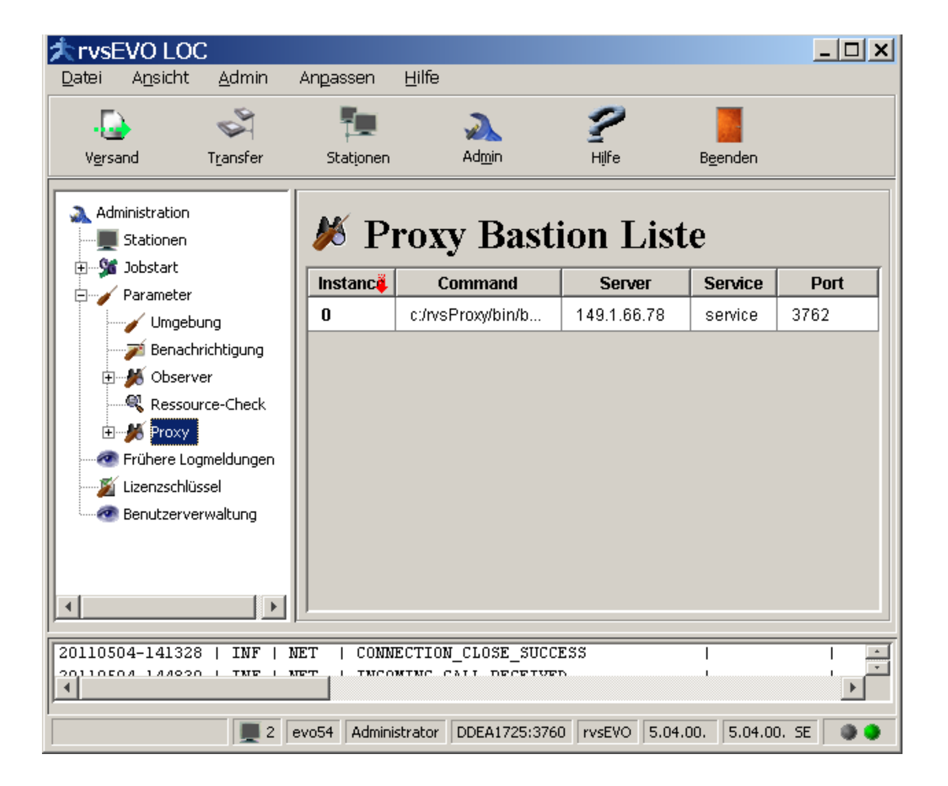
Wir senden Ihnen die Software gerne auch auf DVD zu.

Die Installation der Bastion Instance nehmen Sie bitte, wie im „Benutzerhandbuch rvs® OFTP Proxy“ erläutert, vor.

### 7.3 Bastion Instance konfigurieren

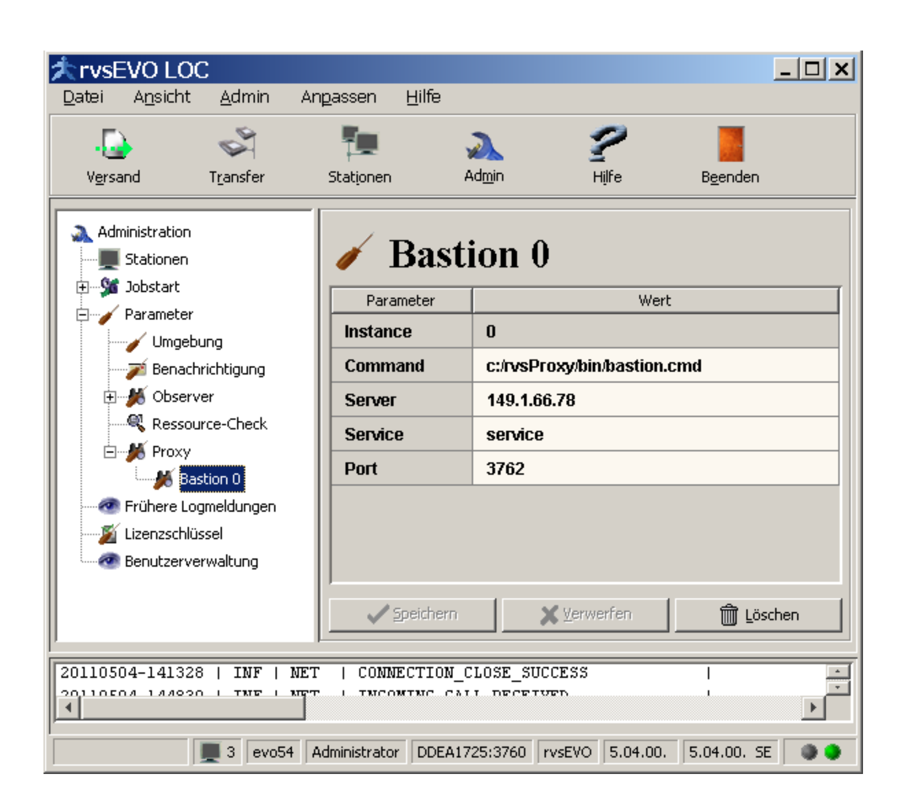
Um eine Verbindung mit der Bastion Instance herstellen zu können, ist es erforderlich, die Parameter in rvsEVO zu hinterlegen und einen Proxy Listener zu konfigurieren.

In der grafischen Benutzeroberfläche (`Admin -> Parameter -> Proxy`) können Sie sich eine Liste der bereits angelegten Bastion Instances anzeigen lassen.




**Bastion konfigurieren**

Um eine Bastion Instance hinzuzufügen, klicken Sie mit der rechten Maustaste auf `Proxy` und wählen anschließend `add proxy`.
Die daraufhin einzugebenen Parameter wurden bereits bei der Installation der Bastion Instance festgelegt. Sollte eine Änderung dieser Werte erforderlich sein, z.B. nach einem Wechsel der Hardware, ist darauf zu achten, dass diese Parameter sowohl im Start Skript `rvsProxy/bin/bastion.cmd (rvsProxy/bin/bastion.sh)` als auch bei der entsprechenden Bastion Instance in rvsEVO angepasst werden.




In folgender Tabelle finden Sie eine Erläuterung der einzelnen Parameter.

Parameter

| Parameter | Beschreibung |
|---|---|
| **Instance** | Fortlaufende Nummer, die von rvsEVO für jede Bastion Instance automatisch vergeben wird. |
| **Command** | Verzeichnis, in welches die Bastion Instance von rvs® OFTP Proxy installiert wurde |
| **Server** | Rechnername oder IP-Adresse des Rechners, auf dem die Bastion Instance installiert wurde |
| **Service** | Name des Bastion Service. Standard: `service`. |
| **Port** | Port der Bastion Instance für die RMI-Kommunikation, auf dem ein Listener gestartet wird. |

Proxy Listener konfigurieren

Für jede Bastion Instance muss mindestens ein Empfänger / Listener definiert werden. Bitte gehen Sie hierzu vor, wie im Kapitel 3.2.2 "Lokale Station konfigurieren" beschrieben.

Ebenso muss eine Nachbarstation, mit der Sie über rvs® OFTP Proxy kommunizieren möchten, mit Proxy TCP/IP bzw. Proxy TLS als Netzwerk

hinzugefügt werden. Nähere Hinweise hierzu finden Sie im Kapitel 3.2.3 "Nachbarknoten hinzufügen und konfigurieren".

Verbindung aufrechterhalten

**Hinweis:** Durch den Versand von Heartbeats von der Bastion Instance zu rvsEVO, kann ein Abbruch der Verbindung nach längerer Übertragungspause verhindert werden. Diese Funktion ist in rvs® OFTP Proxy in der Datei `$RVSPROXY_HOME\conf\RemoteListenerProperties.properties` mittels der Parameter `listener.controller-thread.sleep-time` und `listener.controller-thread.keep-alive-counter` konfigurierbar.
Über den Parameter `listener.controller-thread.sleep-time` wird definiert, in welchen Abständen (in Millisekunden) überprüft werden soll, ob der Remote-Listener noch betriebsbereit ist.
Im Parameter `listener.controller-thread.keep-alive-counter` wird definiert, wie oft eine solche Überprüfung stattfinden soll, bis ein Heartbeat versendet wird.

# 8 File Service Modul

## 8.1 Grundlagen

Das Internet wird zunehmend genutzt, um auch kommerzielle und unternehmenskritische Daten beliebigen Inhalts zu übertragen. Dafür sind solche Vorteile ausschlaggebend wie seine große Verbreitung, die enormen verfügbaren Bandbreiten und die günstigen Übertragungskosten.

Diesen Vorzügen stehen ernst zu nehmende Gefahren gegenüber: Im Internet sind alle Daten Angriffen auf ihre Vertraulichkeit und Integrität ausgesetzt. Ihre Authentizität ist ebenfalls nicht ohne weiteres sichergestellt. Das kann dazu führen, dass Daten von ihrem augenscheinlichen Absender verleugnet werden (repudiation). Solchen Gefahren wird mit geeigneten Protokollen, wie TLS und HTTPS begegnet. Für CAD Daten, wie auch EDI- und beliebige andere Daten ist die OFTP Version 2 als sicheres Internet Protokoll standardisiert worden.

Ein weiteres Risiko besteht durch Malware, die im Internet in großem Umfang und automatisiert verbreitet wird. Viren, Würmer, Trojaner und andere schädliche Programme können beträchtliche Schäden in der IT-Infrastruktur von Unternehmen anrichten. Um dieses Risiko zu minimieren, werden Netzbereiche geschaffen, die von den Unternehmensnetzen durch Firewalls getrennt sind (DMZ). In den DMZ werden beispielsweise http Proxys eingesetzt, die Nutzdaten transportieren, ohne direkte Verbindungen zwischen dem Internet und dem Intranet zuzulassen. Eine ähnliche Lösung wird für OFTP angestrebt.

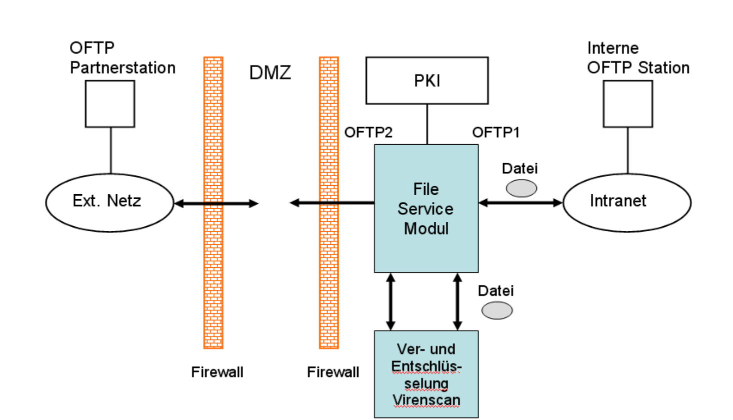
Das File Service Modul wurde entwickelt, um die Sicherheit im Zusammenhang mit der Datenübertragung im Internet zu erhöhen. Aufgabe des File Service Modul ist es, als OFTP-Router  Dateien entsprechend OFTP2 für Stationen zu ver- und entschlüsseln, die selbst OFTP2 nicht unterstützen. Daneben ist das File Service Modul in der Lage, aus dem Internet empfangene Dateien nach der Entschlüsselung auf Viren zu überprüfen.

Das File Service Modul arbeitet auf der Ebene der File Services von OFTP und benötigt deshalb den kompletten Umfang an Features, den ein ‚normaler' OFTP Router zur Verfügung stellt. Deshalb wird das File Service Modul als rvsEVO-Installation betrachtet, die zusätzliche Features enthält.

## 8.2 Architektur des File Service Modul

Das File Service Modul arbeitet als OFTP Router und  ver- und entschlüsselt Dateien stellvertretend für interne Stationen, bevor er sie weiterleitet.

Die folgende Abbildung soll die Architektur des File Service Moduls veranschaulichen:




Folgende Funktionalitäten stehen zur Verfügung:

– Das File Service Modul arbeitet als OFTP-Router

– Protoll-Umsetzung OFTP1 <-> OFTP2, alle Protokollelemente, auch EERP

– Datei- Ver- und Entschlüsselung stellvertretend für interne Stationen

– Virenscan empfangener Dateien

– Verbindung zur PKI für Download von Zertifikaten, OCSP-Prüfung, Download von CRL

## 8.3 Konfiguration einer Nachbarstation bei Anwendung der Komponente File Service Modul

In diesem Kapitel werden die Konfigurationseinstellungen für das File Service Modul  beschrieben.

Eine genaue Beschreibung der Stationskonfiguration finden Sie in Kapitel 3.2 "Anpassen der Stationskonfiguration".

Die folgenden Parameter sind für das File Service Modul relevant:

| | |
|---|---|
| File Service Proxy | Mit diesem Paramter legen Sie fest, ob das File Service Modul aktiviert werden soll. Mögliche Werte:<br>– kein: Funktionalität wird nicht benötigt (Standard)<br>– Intern: diese Partnerstation ist eine interne (im Intranet befindliche) Station; beim Empfang einer verschlüsselten Datei, die weiter an eine interne Station geroutet wird, wird der Entschlüsselungsmechanismus im File Service Modul automatisch aktiviert, je nachdem welche Verschlüsselungskomponenten die Sendepartnerstation verwendet hat.<br>– Extern: diese Partnerstation wird als eine externe (über das Internet erreichbare) Station markiert. Es stehen alle Komprimierungs- und Verschlüsselungsfeatures zur Verfügung. |

Die nachstehenden Parameter sind Komprimierungs- und Verschlüsselungsfeatures, die nur zur Verfügung stehen, wenn Parameter **FileProxy Service** = Extern. Eine Beschreibung der Parameter finden Sie in der Tabelle “Sendeparameter” auf Seite 103

– Sicherheitsmerkmale

– Komprimierung

– Verschlüsselung

– Signatur

– ERP-Signatur

– Verschlüsselungsalgorithmus

**Hinweis:** Es können sowohl direkte Nachbarstationen als auch Routingstationen mit File Service Modul gekennzeichnet werden. Die konfigurierte File Service Modul Funktion ist damit stationsweit festgelegt und kann nicht beim Erstellen eines Sendeauftrags überschrieben werden. Dies entspricht bei eingeschalteter Verschlüsselung einer Zwangsverschlüsselung (**Security**=FORCED).

# 9 Remote GUI

In rvsEVO ab der Version 5.2 ist es möglich, sich mit einem entfernten (remote) Zugriff auf die rvsEVO-GUI einzuloggen.

Voraussetzungen

Folgende Voraussetzungen müssen für eine Remoteverbindung erfüllt sein:

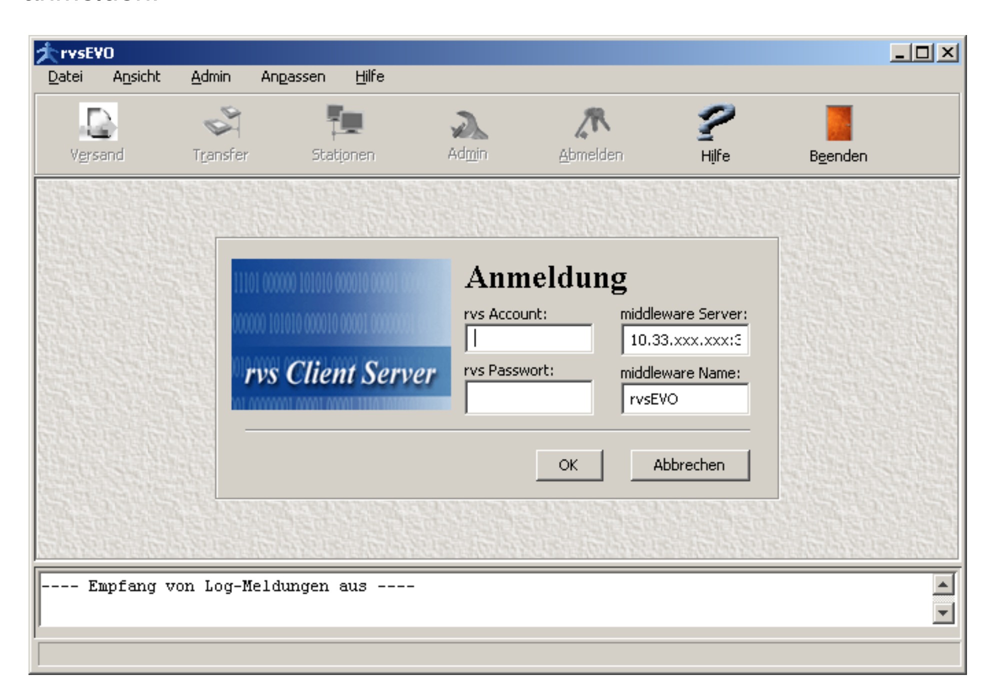
- Auf der Client-Seite muss ein rvsEVO Client installiert werden (siehe Kapitel 2.3 "Neuinstallation von rvsEVO").
- Auf der Server-Seite ist der Client User als Benutzer anzulegen (siehe Kapitel 10.1.1 "Benutzer hinzufügen")
- Der Server muss gestartet sein

## 9.1 Starten der GUI vom entfernten Rechner

Sie können rvsEVO mit Hilfe der rvsEVO-Programmgruppe starten: `Start -> Alle Programme -> rvsEVO -> rvs GUI`.

Alternativ kann diese Funktion auf der Kommandozeile mit dem Skipt `$RVS_HOME\bin\rvsGUI -r -u <user> -p <password>` gestartet werden.

Folgendes Fenster öffnet sich und Sie können sich auf dem Server anmelden:




Erforderliche Angaben:

- `rvs Account`: Login des in rvsEVO (auf dem Server) mit der Benutzerverwaltung eingerichteten Benutzers (Benutzerkennung)

- rvs Passwort: Passwort des Benutzers
- middleware Server: Name oder IP-Adresse des entfernten Rechners und der Port, auf dem der Rechner horcht. Folgende Syntax ist dabei zu beachten: <RMI Server>:<Port>. Als StandardPort ist 3755 zu nehmen.
- middleware Name: Standard rvsEVO, diesen Parameter finden Sie auch in der rvsEVO-Parameter-Liste (Admin -> Param).

## 9.2 Besonderheiten der Remote GUI

Die rvsEVO-GUI auf dem Client-Rechner unterscheidet sich nur unwesentlich von der rvsEVO-GUI auf dem Server-Rechner.

Folgende Unterschiede sind vorhanden:

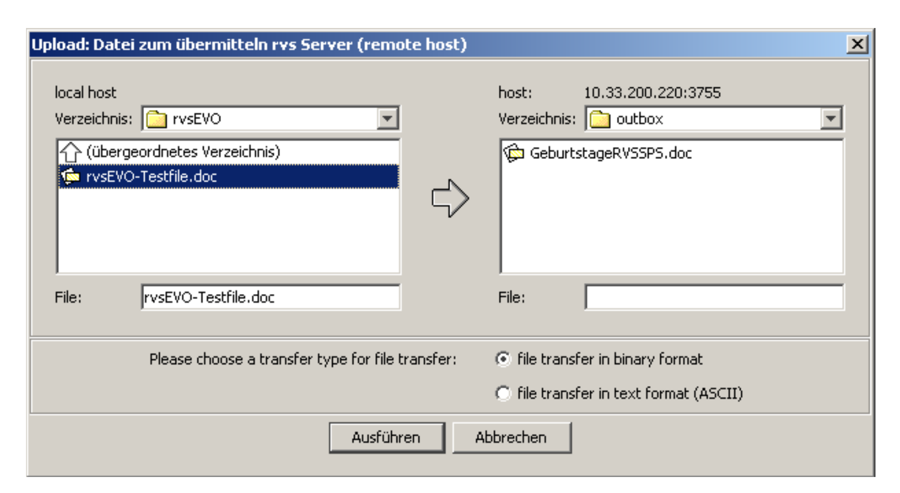
– Die Zertifikatsverwaltung steht nur eingeschränkt zur Verfügung. Die Funktionen **Zertifikat exportieren**, **Zertifikat zu ComSecure exportieren** und **ComSecure öffentlichen Schlüssel importieren** können über die Remote GUI nicht ausgeführt werden.
Bei TLS-Verschlüsselung können die beiden Schlüsselverwaltungsdateien tlsKeyManagerKeyStore.p12 und tlsTrustManagerKeyStore.p12 nicht mit dem Programm Portecle geöffnet werden.

– Im Transfer-Fenster haben Sie die Wahl, ob Sie den Sendeauftrag remote oder lokal erstellen möchten. Dabei sind die Parameter für den Dateiversand gleich geblieben.



– Weiter besteht die Möglichkeit zum Uploading (Dateien zum Server übertragen) oder Downloading (Dateien vom Server holen/ übertragen) von Dateien.

Uploading: Um eine Datei zum Server zu übertragen, wählen Sie im Transfer-Baum den Menüpunkt **uploading.** Das folgende Fenster öffnet sich:




Auf der linken Seite können Sie die zu versendende Datei auswählen. Im Feld `File` auf der rechten Seite können Sie der Datei einen neuen Namen zuweisen. Im unteren Bereich haben Sie die Möglichkeit, das Format der zu versendenden Datei anzugeben.

Mit `Ausführen` bestätigen Sie Ihre Eingaben und starten die Übertragung. Auf dem rvsEVO Server wird die Datei im Verzeichnis `$RVS_HOME\files\outbox gespeichert.`

**Hinweis**: Für einen Upload öffnet der Client einen Port über den sich der Server verbinden kann. Dieser ist über die Definition `<entry key= "client.remotefileloader.port" value="3756" />` in der Datei $RVS_HOME`/conf/rvsEvoClient.prefs` konfigurierbar.

Downloading Um eine Datei vom Server zu empfangen, wählen Sie im Transfer-Baum den Menüpunkt **downloading**.

Im rechten Bereich des Fensters werden die auf dem Server (im Verzeichnis `$RVS_HOME\files\inbox`) zum Download bereitgestellten Dateien aufgelistet. Wählen Sie die Datei aus, die Sie herunterladen möchten. Auf der linken Seite des Fensters können Sie das Verzeichnis auswählen, in dem die Datei abgelegt werden soll und im Feld `File` der Datei einen neuen Namen zuweisen.

Mit `Ausführen` bestätigen Sie Ihre Eingaben und starten die Übertragung.

Kommando-zeilentools Bei allen Kommandozeilenaufrufen, die remote gestartet werden, muss die Benutzerkennung und das Passwort, des mit der Benutzerverwaltung eingerichteten Benutzers, angegeben werden.

Beispiel:

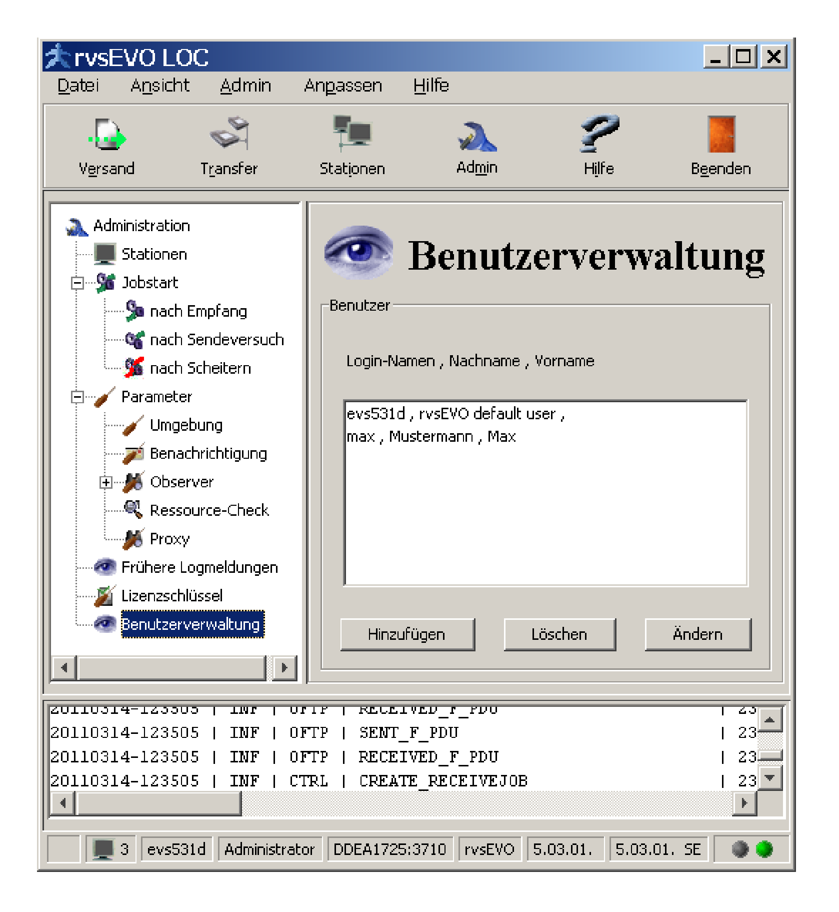
```
updateStationList -f <xml-filename> -u <user> -p
<password>
```

# 10 Benutzerverwaltung

Die Funktionalität Benutzerverwaltung steht ab der rvsEVO-Version 5.2 zur Verfügung. Sie ist über das Symbol **Admin** in der GUI-Funktionsleiste (`Admin -> User Managment`) erreichbar.

## 10.1 Verwaltung eines Benutzers

Wenn Sie das Symbol für Benutzervewaltung anklicken, erhalten Sie auf der rechten Seite das Fenster für die Verwaltung der einzelnen Benutzer.

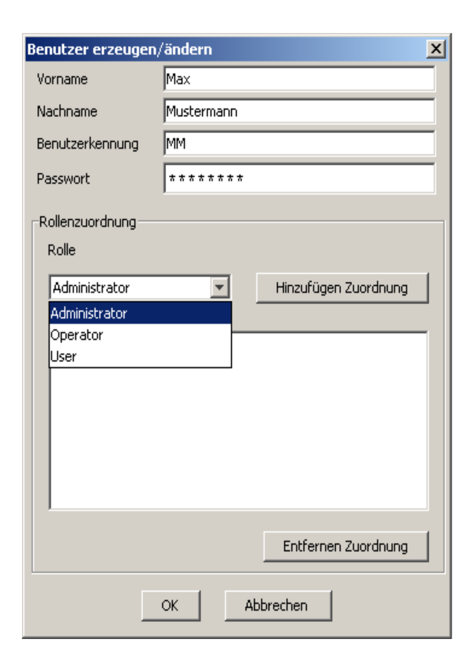



Folgende Operationen sind im Benutzerfenster möglich:

- Hinzufügen: einen neuen Benutzer anlegen
- Löschen: einen vorhandenen Benutzer deaktivieren/löschen
- Ändern: einen vorhandenen Benutzer editieren

### 10.1.1 Benutzer hinzufügen

Wenn Sie einen neuen Benutzer hinzufügen möchten, klicken Sie auf die Schaltfläche **Hinzufügen**. Es öffnet sich die Maske Benutzer erzeugen/ändern.




In dieser Maske können Sie folgende Parameter eintragen:

- Vorname (des Benutzers)
- Nachnahme (des Benutzers)
- Benutzerkennung (für die Anmeldung)
- Passwort (für die Anmeldung)
- Rolle (Administrator, Operater oder User, auswählbar über die Combo-Box, aktivieren über die Schaltfläche **Hinzufügen Zuordnung**).

Diese Eingaben müssen mit **OK** bestätigt werden.

**Hinweis**: Ein einfacher **Benutzer** darf das Programm rvsEVO zum Senden und Empfangen von Dateien benutzen. Ein **Operator** hat Benutzerrechte und darf zusätzlich Operator-Kommandos ausführen. Ein **Administrator**, darf darüber hinaus das  rvsEVO-System konfigurieren.

### 10.1.2 Benutzer löschen

Um einen Benutzer zu löschen, markieren Sie bitte den zu löschenden Benutzer. Anschließend kann der Benutzer über die Schaltfläche **Löschen** entfernt werden.

### 10.1.3 Benutzer ändern

Zur Bearbeitung der Daten eines Benutzers markieren Sie bitte den Benutzernamen und gehen Sie über die Schaltfläche **Ändern** in die Maske Benutzer erzeugen/ändern. Die angezeigten Benutzerdaten können editiert werden.
Zur Änderung der Rollenzuordnung muss zuerst die bisherige Rolle markiert und über die Schaltfläche **Entfernen Zuordnung** gelöscht werden. Anschließend kann eine neue Rolle zugewiesen werden.

Die Eingaben müssen mit **OK** bestätigt werden.

**Achtung:** Die Daten des Standardbenutzers können nicht editiert werden.

# 11 PKI-Anbindung mit rvsEVO

In diesem Kapitel wird die Konfiguration und die Funktionalität der PKI-Anbindung in rvsEVO beschrieben.

## 11.1 Einführung

In einer PKI (Public Key Infrastructure) werden öffentliche Schlüsssel verwaltet und Zertifikate für sie ausgestellt. Der Zugriff auf Zertifikate in einer PKI erfolgt bei rvsEVO über einen LDAP-Verzeichnisdienst (rvsEVO unterstützt LDAP version 3).

**Hinweis**: LDAP (Leightweight Directory Access Protocol) ist ein Netzwerkprotokoll, welches die Kommunikation zwischen einem LDAP-Client und einem LDAP-DirectoryServer (Verzeichnis-Server) regelt. Das Protokoll bietet folgende Funktionen: Anmeldung vom Client am Server, Suchabfrage bzgl. der im Verzeichnis (directory) gespeicherten Informationen und Modifikation von Informationen.

Folgende Funktionalitäten bietet die rvsEVO-PKI-Anbindung:

– Beziehen von Zertifikaten für Partnerstationen via LDAP-Schnittstelle von einer PKI.
– Prüfen der Zertifikate mit OCSP (Online Certificate Status Protocol), CRL oder CertPath
– Übertragung der Informationen zum OCSP-Server mit HTTP

## 11.2 Konfiguration

Folgende Schritte sind durchzuführen, um die PKI-Anbindung in rvsEVO zu konfigurieren:

– Konfiguration der Stationen, die an Stelle der lokalen Schlüsselverwaltungsdatei eine PKI verwenden in der GUI oder in der Stationsliste `$RVS_HOME/conf/rvsStationlist.xml` (siehe Kapitel 11.2.1).
– Für den PKI-Zugriff müssen die Zugriffsparameter in der Konfigurationsdatei `$RVS_HOME/conf/PkiParameter.xml` konfiguriert werden.

### 11.2.1 Konfiguration der Stationsliste

Die PKI-Anbindung wird in rvsEVO stationenbezogen konfiguriert.

In der XML-Datei `$RVS_HOME/conf/rvsStationlist.xml` wird die PKI-Anbindung mit dem Element `<Pki>` innerhalb des XML-Elements `<StationNeighbour>` oder/und innerhalb des Elements `<LocalStation>` konfiguriert.

Für Informationen über die Konfiguration mittels der grafischen Benutzeroberfläche (GUI) sehen Sie bitte Kapitel 3.2.1 „Konfiguration von Stationen mittel GUI".

**Hinweis**: Im Normalfall wird die PKI-Anbindung für Partnerstationen eingerichtet. Für die lokale Station ist eine PKI-Anbindung nur dann interessant, wenn man das eigene Zertifikat validieren möchte.

**Beispiel** (`rvsStationlist.xml`, XML-Element `<Pki>`):

```
<Pki>
    <CertificateValidationType>NONE</CertificateValidationType>
    <PkiEnabled>true</PkiEnabled>
</Pki>
```

In diesem Beispiel wird für eine Partnerstation die PKI-Anbindung ohne Zertifikatsvalidierung eingeschaltet.

Für das XML-Element `<Pki>` sind Unterelemente `<PkiEnabled>` und `<CertificateValidationType>` möglich.

| Parameter | Beschreibung |
|---|---|
| **<CertificateValidationType>** | Dieses XML-Element beschreibt die Art der Zertifikatsvalidierung. Es werden 3 unterschiedliche Validierungs-Methoden unterstützt: `OCSP` (Online Certificate Status Protocol), `CRL` (Certificate Revocation List) und `CertPath`.<br>`None`: Der Wert `None` im XML-Element `<CertificateValidationType>` bedeutet, dass keine Validierung durchgeführt werden muss. |
| **<PkiEnabled>** | Mögliche Werte: `true` und `false`.<br>`true`: aktiviert die PKI-Funktionalität mit LDAP-Verwendung. In diesem Fall muss die XML-Konfigurationsdatei `$RVS_HOME/conf/PkiParameter.xml` eine gültige LDAP-Konfiguration enthalten.<br>`false`: die Zertifikate werden aus der lokalen Schlüsselverwaltungsdatei (Standardwert: `$RVS_HOME/system/data/keystore.p12)` verwendet. |

**Hinweis**:
Wenn als Validierungsmethode `OCSP` oder `CRL` gewählt wurde, muss eine gültige Konfiguration in der Datei `$RVS_HOME/conf/PkiParameter.xml` existieren.

Welche von den drei Validierungs-Methoden eingesetzt wird, hängt in der Regel von der Sicherheitspolitik der Firma ab. Die "stärkste" Prüfung, die rvsEVO anbietet ist OCSP. Danach folgt CRL und anschließend CERT_PATH.

Wenn Sie sich für OCSP entscheiden, muss der OCSP-Service öffentlich (Internet, Intranet) zugänglich sein und Ihre Station muss in der Lage sein auf dieses Netz zuzugreifen.

Die Prüfung gegen CRL kann auch offline erfolgen. Der rvsEVO-Benutzer/Administrator muss die Liste der Zertifikate (CRL) besorgen und sie manuell einspielen. Der Nachteil ist, dass zwischen den Aktualisierungen ein Zertifikat ungültig werden kann, was erst bei der nächsten Aktualisierung festgestellt werden kann.

Die CERT_PATH Validierung benötigt die geringsten Voraussetzungen. Es werden lediglich die ROOT-Zertifikate und/oder die Zertifikate der CA's benötigt. Diese Zertifikate ändern sich in der Regel sehr selten (mehrere Jahre), so dass ein sehr geringer Wartungsaufwand besteht. Bei der Validierung wird aber nur geprüft, ob das Zertifikat formal richtig ist, d.h. alle Signaturen in Ordnung (nicht gefälscht) sind, das Zertifikat nicht abgelaufen ist (Datum gültig), der Zertifikatsaussteller in Ordnung ist usw. Dieser Vorgang wird anschließend mit dem Zertifikatsaussteller (issuer) ausgeführt (und mit dessen Zertifikatsaussteller usw.). Beendet ist die Prüfung, wenn ein vertrauenswürdiges Zertifikat (i.d.R. root Zertifikat) als Zertifikatsaussteller gefunden wurde. Es wird dabei nicht geprüft, ob ein Zertifikat zurückgezogen (revoziert) wurde (z.B.: wenn der öffentliche Schlüssel oder der private Schlüssel kompromitiert wurde, das Zertifikat nicht mehr benötigt wird, sich der Name geändert hat usw.).

### 11.2.2 PKI-Konfigurationsdatei

Die PKI-Konfiguration wird in der Datei `$RVS_HOME/conf/PkiParameter.xml` durchgeführt.

**Beispiel**:

```
<pkiParameter>
<pkiProfile>
<pkiProfileId>default</pkiProfileId>
   <!-- =================== -->
   <!--   LDAP Einstellungen -->
   <!-- ========================= -->
   <ldapParameter>
     <!-- ================ -->
     <!-- ldap server info -->
     <!-- ================ -->
     <ldapServer>
       <!-- server name or ip-address -->
       <name>localhost</name>
       <!-- ip-port -->
       <port>10389</port>
       <!-- LDAP version to use -->
       <version/>
     </ldapServer>
     <!-- ================ -->
     <!-- access info      -->
     <!-- ================ -->
     <ldapSecurity>
       <user/>
       <password/>
```

```
    </ldapSecurity>
    <!-- ============================================================ -->
    <!-- search pattern definition, how to retrieve X.509 Certificates -->
    <!-- ============================================================= -->
    <ldapCertificateSearchPattern>
     <!--  root node (distinguishedName) in the X.500 directory,
      where search start from -->
      <root>ou=Prozesse,o=Volkswagen AG,dc=VW,dc=vwg,dc=com</root>
      <!-- common name pattern where to find the right certificate -->
      <commonName>cn=VW_OFTP </commonName>
      <!-- attribute which includes the X.509 certificate-->
      <certificateAttribute>userCertificate</certificateAttribute>
      <reverseOrder>false</reverseOrder>
    </ldapCertificateSearchPattern>
  </ldapParameter>
 <!-- ============================== -->
 <!--         OCSP Einstellungen     -->
 <!-- ============================== -->
  <ocspParameter>
    <!-- URI of the OCSP Service -->
   <accessLocation>http://localhost:8080/ejb/publicweb/status/ocsp</accessLocation>
    <!-- ocspResponderDistinguishedName>CN=rvsRootCA,OU=rvs,O=tsystems
     </ocspResponderDistinguishedName-->
     <!-- issuerDistinguishedName></issuerDistinguishedName-->
     <!-- defines the handling of OCSP-failures -->
    <!--  ignoreMissingServer Define: if it should be ignored, when OCSP server is
      not reachable boolean value (true/false) if set to true, OCSP check returns
      VALID, when certificate will be checked, but OCSP server could not be
       connected -->
     <ignoreMissingServer></ignoreMissingServer>
     <!-- if set true, OCSP response signature and responder certificate will not be
       verified -->
     <skipOcspResponseValidation></skipOcspResponseValidation>
   </ocspFailureHandling>
  </ocspParameter>
  <!-- =================== -->
  <!--   CRL Einstellungen -->
  <!-- =================== -->
  <crlParameter>
    <!-- directory where the actual used CRLs are stored -->
    <location>g:\CRL\</location>
    <url/>
    <!-- defines the handling of CRL-failures -->
    <crlFailureHandling>
    <!--- if set true, CRL and CRL issuer are not verified -->
    <skipCRLValidation>false</skipCRLValidation>
   </crlFailureHandling>
  </crlParameter>
</pkiProfile>
</pkiParameter>
```

Die folgende Tabelle enthält die Erklärungen der Parameter aus der Datei `$RVS_HOME/conf/PkiParameter.xml`.

| Parameter | Beschreibung |
|---|---|
| **accessLocation** | URL des OCSP-Service. |
| **certificateAttribute** | definiert das Attribut, in dem das Benutzerzertifikat gespeichert ist (z.B. userCertificate, userCertificate;binary). |
| **commonName** | eindeutiger Name, der ein Zertifikat innerhalb des Basisverzeichnisses von den anderen Zertifikaten unterscheidet. In der ersten Version wird dieser Wert immer um die ODETTE-ID der entsprechenden Station erweitert. Der endgültige DN setzt sich demnach wie folgt zusammen: `<root>` + `<commonName>` + `ODETTE-ID`. |
| **crlFailureHandling** | beschreibt das Verhalten bei fehlerhaften CRL-Überprüfungen. |
| **crlParameter** | beschreibt Parameter für die Zertifikatsprüfung per CRL. |
| **cspFailureHandling** | Beschreibt das Verhalten bei fehlerhaften OCSP-Anfragen. |
| **ignoreMissingServer** | Mögliche Werte: `true` und `false`.<br>`true`: Bei Fehlern während einer OCSP-Anfrage wird das Zertifikat als gültig erklärt, auch wenn z.B. der OCSP-Service nicht erreicht werden kann.<br>`false`: Wenn die OCSP-Anfrage fehlschlägt, wird das Zertifikat als nicht gültig erklärt. |
| **issuerDistinguished-Name** | Optional: Erlaubt die Angabe des DN des Zertifikatsausstellers (issuer). In der Regel wird der Zertifikatsaussteller im Zertifikat angegeben. Für Sonderfälle kann hier ein anderer Name angegeben werden. |
| **ldapCertificateSearch-Pattern** | beschreibt Regeln für den Aufbau des DN (distinguished name) eines Zertifikats innerhalb eines LDAP-Verzeichnisses. |
| **ldapParameter** | In dieser Gruppe werden die Parameter für den Zugriff auf das LDAP-Verzeichnis beschrieben. |
| **ldapSecurity** | Parameter für LDAP-Authentifizierung (es wird nur einfache Authentifizierung unterstützt). Werden die Elemente nicht definiert, wird die Authentifizierung übersprungen (Server muss `ANONYMOUS`-Authentifizierung unterstützen). |

| | |
|---|---|
| **ldapServer** | IP-Konfiguration des LDAP-Servers. |
| **location** | gibt das Verzeichnis an, in dem die CRLs gespeichert sind. Die CRLs müssen vom Benutzer selbst gepflegt werden und auf dem neusten Stand gehalten werden. |
| **name** | IP-Adresse oder DNS-Name des LDAP-Servers. |
| **ocspParameter** | definiert Parameter für die Zertifikatsprüfung per OCSP. |
| **ocspResponderDistinguishedName** | DN des OCSP-Responder-Zertifikats (optional). Über den DN kann das OCSP-Responder-Zertifikat gesucht werden. Die Angabe des DN ist z.B. nicht notwendig, wenn das Responder-Zertifikat mit dem OCSP-Response übertragen wird. In anderen Fällen unterschreibt z.B. der Zertifikatsaussteller auch den OCSP-Response. Der DN des Zertifikatsausstellers (issuer) ist häufig Bestandteil des Zertifikats. |
| **password** | Passwort |
| **port** | IP-Port des LDAP-Servers. |
| **reverseOrder** | gibt die Reihenfolge des DN (distinguished name) an.<br>Standard: `true`. Bei einigen LDAP-Verzeichnissen muss der CN (commonName) vor oder hinter `<root>` angegeben werden.<br>`true:` folgende Reihenfolge wird verwendet<br>`<commonName>`+ `ODETTE-ID`+`<root>`.<br>`false:` es wird die Reihenfolge wie folgt verwendet:<br>`<root>`+ `<commonName>`+ `ODETTE-ID`. |
| **root** | LDAP-Basisverzeichnis, innerhalb dessen alle Zertifikate gefunden werden können. |
| **skipCrlValidation** | eine CRL muss auf ihre Gültigkeit geprüft werden. Dazu gehört auch die CRL-Signatur. Mögliche Werte: `true` und `false`.<br>`true`: Die Überprüfung wird übersprungen.<br>`false`: Die Überprüfung wird nicht übersprungen.<br>**Hinweis**: Das Überprüfen der Signatur beinhaltet auch die Prüfung des CRL-issuer-Zertifikats per CERT_PATH. |

| | |
|---|---|
| **skipOcspResponseValidation** | Ein OCSP-Response muss auf seine Gültigkeit überprüft werden. Dazu gehört auch die Überprüfung der OCSP-Response-Signatur.<br>Mögliche Werte: `true` und `false`.<br>`true`: Die Überprüfung wird übersprungen.<br>`false`: Die Überprüfung wird nicht übersprungen.<br>**Hinweis**: Das Überprüfen der Signatur beinhaltet auch die Überprüfung des Responder-Zertifikats per CERT_PATH. |
| **user** | Benutzername |

# 12 rvsEVO-Datenbank

In rvsEVO ab der Version 5.0 ist es möglich, rvsEVO-Jobdaten in eine Datenbank anstatt ins Verzeichnis `$RVS_HOME/jobs` mit den Unterverzeichnissen `ENDED`, `FAILED`, `SND` oder `RCV` zu schreiben.

Ab der Version 5.02 werden auch Daten der Benutzerverwaltung in der rvsEVO-Datenbank gespeichert.

Zurzeit stehen folgende Datenbanken zur Verfügung: derby Embedded oder Oracle.

Bei der Installation von rvsEVO werden Sie gefragt, ob Sie als Persistenzmechanismus **derby** oder **Oracle** wünschen (siehe Kapitel 2.3 "Neuinstallation von rvsEVO").

## 12.1 Derby

Derby-Datenbank ist eine lizenzfreie, Java-basierte relationale Datenbank von der Apache Foundation.

Für Derby Embedded gilt, dass eine Derby-Datenbank von rvsEVO mitgebracht und von der Installationsroutine im lokalen Verzeichnis `$RVS_HOME/db` installiert wird.

Folgende Derby-Datenbanken wurden getestet:

– Derby 10.3.1.4

– Derby 10.2.2.0

## 12.2 Oracle

Wenn Sie sich für eine Oracle-Datenbank entschieden haben, müssen folgende Voraussetzungen erfüllt sein, damit rvsEVO erfolgreich installiert werden kann:

– ein Oracle-Datenbankbenutzer wurde mit den Rechten connect, resource, create session und create table angelegt

– die Oracle-Konfiguration wurde einwandfrei eingerichtet. Wenn ein Oracle-Client auf Ihrem Rechner installiert wurde, können Sie die Konfiguration mit folgendem sqlplus-Befehl testen:

```
sqlplus ORACLE-Benutzer@ORACLE-Netzwerkdienstname/
Kennwort
```

**Beispiel**:

```
sqlplus skk@RVS.TSYSTEMS.DE/skk
```

Wenn dieser Befehl erfolgreich war, haben Sie die Sicherheit, dass die Oracle-Konfiguration für rvsEVO einwandfrei eingerichtet ist und können anschließend mit der rvsEVO-Installation beginnen.

Folgende Oracle-Datenbanken wurden getestet:

– Oracle 9i

– Oracle 10g
– Oracle 11g

Im rvsEVO-Installationsdialog (siehe Kapitel 2.3 "Neuinstallation von rvsEVO") werden folgende Oracle-Verbindungsparameter abgefragt:

`jdbc.url`, `jdbc.user`, `jdbc.password` und `Net Service Name`.

`jdbc.url` hat die folgende Syntax:

```
jdbc:oracle:thin:@<server>:<port>/<service_name>
```

**Beispiel**:

```
jdbc:oracle:thin:@localhost:1527/rvsORA
```

Als Standardport für Oracle ist 1521 vorgesehen.

Wenn Ihre Oracle-Datenbank auf einem externen Rechner läuft, muss dieser in der Oracle-Konfigurationsdatei

`$ORACLEHOME/network/admin/tnsnames.ora`

definiert sein.

**Beispiel** (tnsnames.ora):

```
#TNSNAMES.ORA Network Configuration File: /opt/oracle/product/8.1.7/
network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

RVS.TSYSTEMS.DE =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = rvsaix3)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = rvsDB)
    )
  )
```

In diesem Beispiel ist `RVS.TSYSTEMS.DE` ein Oracle-Netzwerkdienstname (`Net Service Name`) und `rvsDB` ein Dienstname (`SERVICE_NAME`). Über einen Netzwerkdienstnamen wird eine Oracle-Datenbank im Netzwerk identifiziert. Dieser Name soll nicht mit dem globalen Datenbanknamen verwechselt werden, auch wenn er ähnlich aufgebaut ist. Mit dem Dienstnamen wird die Zuordnung zu einer Datenbankinstanz gewährleistet, da mehrere Instanzen auf einem Rechner laufen können.

`jdbc.user` ist der Benutzer, der auf der Oracle-Datenbank eingerichtet ist und `jdbc.passsword` ist sein Passwort.

**Hinweis**: Die Verbindungsdaten aus dem Installationsdialog werden in die Datei `$RVS_HOME/conf/jdbc.properties` gespeichert.

**Verbindungspooling (Connection Pooling):**

Unter Verwendung einer Oracle-Datenbank bietet rvsEVO die Funktionalität Connection Pooling an. Darunter wird verstanden, dass ständig Verbindungen zur Datenbank in einem Pool aufrecht erhalten werden. Wenn eine Verbindung zur Datenbank benötigt wird, wird eine vorhandene Verbindung aus dem Pool verwendet, und es muss nicht bei jedem Zugriff eine neue Verbindung aufgebaut werden. rvsEVO verwendet C3P0 (Hibernate) für Connection Pooling.

Um Firewall-Problemen vorzubeugen, sollte die Datei $EVO_HOME/system/data/Oracle/hibernate.cfg.xml um C3P0-Parameter ergänzt werden.

Beispiel für eine Grundkonfiguration:

```
<property
name="connection.provider_class">org.hibernate.connection.C3P0Con
nectionProvider</property>
<property name="hibernate.c3p0.idle_test_period">300</property>
<property name="hibernate.c3p0.max_size">10</property>
<property name="hibernate.c3p0.min_size">0</property>
```

Der wichtigste Parameter ist hibernate.c3p0.idle_test_period. Dieser Parameter bestimmt, wie oft (Zeitangabe in Sekunden) der Connection Pool auf inaktive / aktive Verbindungen überprüft wird. Mit dem Parameter max_size können Sie die maximale Anzahl und mit min_size die Anfangszahl der Datenbankverbindungen festlegen.

Weitere Informationen entnehmen Sie bitte der C3P0-Dokumentation:

http://community.jboss.org/wiki/HowToconfiguretheC3P0connectionpool

## 12.3 Datenbanktabellen löschen und neu anlegen

**Hinweis**: Stoppen Sie rvsEVO bevor Sie Datenbank-Skripte ausführen und starten Sie es wieder neu, wenn Sie eine neue Datenbank angelegt haben.

Wenn die alten Job- und Benutzerdaten nicht mehr benötigt werden, können Sie mit dem Skript $RVS_HOME/system/DERBY/derby_drop oder $RVS_HOME/system/ORACLE/oracle_drop/ rvsEVO-Datenbanktabellen löschen. Anschließend müssen die Skripte $RVS_HOME/system/DERBY/derby_create und $RVS_HOME/system/DERBY/derby_user für derby oder $RVS_HOME/system/ORACLE/oracle_create und $RVS_HOME/system/DERBY/oracle_user für Oracle ausgeführt werden, um die Datenbanktabellen neu anzulegen.

### 12.4 Jobdaten aus der Datenbank anzeigen

Der sql-Befehl für das Anzeigen der Daten aus einer Datenbanktabelle lautet:

```
SELECT * from <TABLE>;
```

Für die Jobdaten-Tabelle `HIB_JOB_DATA` lautet der Befehl zum Anzeigen aller Jobdaten:

```
SELECT * from HIB_JOB_DATA;
```

Wenn Sie nur einen Jobdataparameter anzeigen möchten (z.B. `JOBID`) muss der folgende Befehl angewendet werden.

```
SELECT JOBID from HIB_JOB_DATA;
```

**Hinweis**: Bei Derby Embedded ist es leider nicht möglich, sich während des Betriebs die Jobdaten anzeigen zu lassen, da die rvsEVO-Applikation auf die Datenbank zugreift.

# 13 Zentrale Administration von rvsEVO

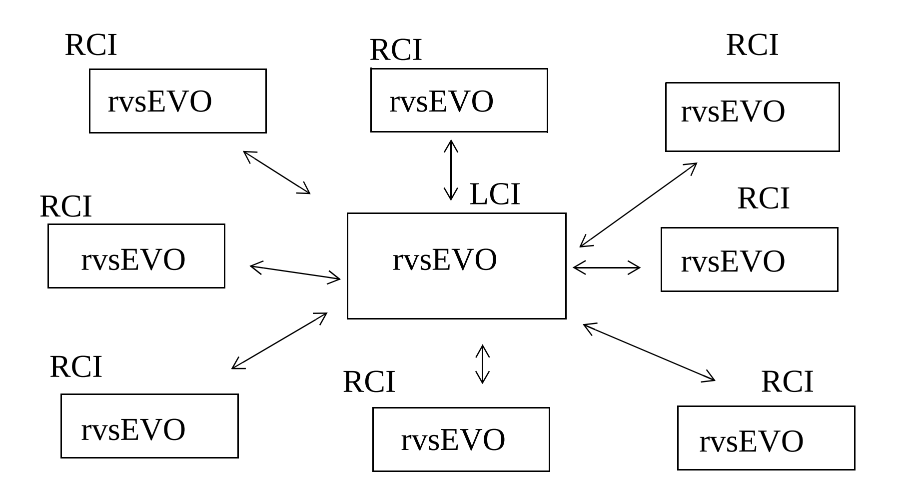
In diesem Kapitel ist eine mächtige Funktionalität von rvsEVO beschrieben: wie man mit einer rvsEVO-Installation andere rvsEVO Stationen administrieren kann. Die Zentrale Administration ist von besonderer Bedeutung für den rvsEVO-Netzwerk-Administrator, da sie ihm ermöglicht, alle rvsEVO-Installationen entfernt zu pflegen.

## 13.1 Einleitung

Die Zentrale Administration von rvsEVO ermöglicht das Administrieren von zahlreichen rvsEVO-Installationen mit Hilfe spezieller Konfigurationsdateien.

Die Konfigurationsdateien werden von der rvsEVO-Station gesendet, die andere rvsEVO-Stationen administriert (sie wird in diesem Buch Local Configuration Instance, LCI, genannt) zu der Station, die administriert werden soll (wir nennen sie: Remote Controlled Instance RCI). Die beiden Stationen (Instanzen) müssen in einem Stern-Netzwerk mit einander kommunizieren können (Siehe die folgende Abbildung).

Abbildung




Konfigurations dateien

Die Konfigurationsdateien für die Zentrale Administration nennen wir **Container - Konfigurationsdateien**, um sie von den normalen rvsEVO XML-Konfigurationsdateien wie z.B. `rvsStationlist.xml` zu unterscheiden.

Container - Konfigurationsdateien sind:

– `cfg.req.jar` für Konfigurationsanfragen, die LCI an RCI sendet

– `cfg.rsp.jar` für Konfigurationsantworten, die RCI an LCI sendet.

**Die Local Configuration Instance LCI**

LCI Jede rvsEVO-Installation kann eine LCI werden.

Es gibt ein spezielles Verzeichnis in rvsEVO, wo Konfigurationsdaten von allen RCIs (rvsEVO-Stationen, die entfernt administriert werden sollen) gespeichert werden - das ist das Configuration Repository CRep. Das CRep-Verzeichnis ist `$RVS_HOME/management`. Siehe bitte Kapitel 1.6 für eine detaillierte Erklärung von `$RVS_HOME`.

CRep Das CRep (`$RVS_HOME/management`) besteht aus folgenden Verzeichnissen oder Dateien:

- `$RVS_HOME/management/mgmt-datastore`
  Dieses Verzeichnis entsteht nur nach einer erfolgreichen Übertragung und erhaltenen Antwort auf eine Anforderung (request). Die Anforderung einer Container-Konfigurationsdatei stellt eine LCI an eine RCI. Siehe bitte Kapitel 13.3 für die Erklärung, wie eine Anforderung erzeugt wird.
- `$RVS_HOME/management/mgmt-log/activity.log` Diese log-Datei enthält Protokolle aller Konfigurationsschritte. Jeder Eintrag in dieser Datei besteht aus einem Zeitstempel, dem Typ der Konfiguration: **configuration request** (Konfigurations - Anforderung) oder **configuration response** (Konfigurations - Antwort), der SID der administrierten rvsEVO-Station (RCI) und der Aktivitätsmeldung.
- `$RVS_HOME/management/mgmt-templates`
  Dieses Verzeichnis enthält templates für die Aktionen, die der rvsEVO-Netzwerk-Administrator durchführt. Sie brauchen dieses Verzeichnis nur, wenn Sie die Administrator-Software updaten.
- `$RVS_HOME/management/mgmt-workspace`
  In diesem Verzeichnis sind Konfigurationsdateien von RCIs, die administriert werden sollen, gespeichert. Das Verzeichnis enthält Unterverzeichnisse, die nach der RCIs SID benannt werden (Siehe bitte das Programm `prepareUpdateStation`, Kapitel 13.3 als Beispiel).

**Die Remote Controlled Instance RCI**

RCI Eine RCI ist die rvsEVO-Station, die entfernt administriert werden soll.

Wenn eine rvsEVO-RCI eine Konfigurationsanforderung (Datei `cfg.req.jar`) von einer LCI empfängt, löst dieser Vorgang den Start eines entsprechenden Jobs aus. Dieser Job verarbeitet die Konfigurationsanforderung und generiert eine Konfigurationsantwort. Die Konfigurationsantwort wird dann an die LCI als Datei `cfg.rsp.jar` zurückgesendet.

## 13.2 Programme der Zentralen Administration

Programme

Der ganze Prozess (alle Konfigurations- und Administrationsfälle) sind mit den folgenden Programmen auszuführen. Diese Programme befinden sich im Verzeichnis `$RVS_HOME\bin` als Batch-Dateien (Siehe Kapitel 1.6 für eine detaillierte Erklärung von `$RVS_HOME`). Um sie ausführen zu können, müssen Sie sie aus dem Verzeichnis `$RVS_HOME\bin` starten.

| | |
|---|---|
| orderConfiguration -s <SID> | holt die Konfiguration von einer RCI und speichert diese ins CRep-Verzeichnis.<br><br>**Beispiel** (für CRep):<br>`C:\rvsEVO\management\mgmt-datastore\TINYPW`<br><br>In diesem Beispiel ist `TINYPW` die SID von der RCI (von der Station, die administriert werden soll). |
| prepareUpdateStation -s <SID> | holt eine Kopie der RCI-Konfiguration aus dem CRep-Verzeichnis ins WorkDir, um dort die Konfiguration durchzuführen.<br><br>**Beispiel** für WorkDir:<br><br>Im Verzeichnis `C:\rvsEVO\management\mgmt-workspace` wird folendes Unterverzeichnis erzeugt: `TINYPW\UPDATE_STATION_040826_114418\out`.<br><br>Das ist das Unterverzeichnis für die Konfiguration der Station `TINYPW`; Datum der Erzeugung ist 2004-08-26, Zeit 11:44:18. |
| commitUpdateStation -s <SID> -d <WorkDir> | sendet die modifizierte Konfiguration aus dem WorkDir (ohne Unterverzeichnis `out`) zur RCI (Die RCI soll in der Option `-s` für die StationsID angegeben werden).<br><br>**Beispiel**:<br><br>`commitUpdateStation -s TINYPW -d C:\rvsEVO\management\mgmt-workspace\TINYPW\UPDATE_STATION_040826_114418` |

**Hinweis**: Der Name des Verzeichnisses WorkDir besteht aus dem CRep (Siehe Kapitel 13.1); der SID der RCI (`TINYPW` im obigen Beispiel), dem Verzeichnis `UPDATE_STATION` mit dem Zeitstempel (der wiederum aus Datum und Uhrzeit besteht; im Beispiel `040826_114418`) und dem Verzeichnis `out`.

WorkDir enthält alle wichtigen rvsEVO-Verzeichnisse und Dateien: Hier ist ein Beispiel, wie Sie die Konfigurationsdateien ändern können:

– Ändern Sie die Stationskonfiguration, indem Sie die Datei `$RVS_HOME/conf/rvsStationlist.xml` editieren.
– Ändern Sie die Jobstarts-Konfiguration, indem Sie die Datei `$RVS_HOME/conf/rvsJobstart.xml` editieren.
– Machen Sie ein Update der Software, indem Sie die .jar-Dateien im Verzeichnis `$RVS_HOME/lib` austauschen.

### 13.3 Wie arbeite ich mit der Zentralen Administration?

Die folgenden Schritte sollten Ihnen als ein typisches Konfigurationsbeispiel dienen. Alle diese Schritte sind notwendig, unabhängig davon, ob Sie ein Update durchführen, einen Lizenzschlüssel austauschen oder die Konfiguration der Stationen, Parameter oder Jobstarts ändern möchten. Mehr Details über derartige Administrationsaufgaben erhalten Sie in den Kapiteln 13.3.1, 13.3.2 und 13.3.3.

**Hinweis**: Die folgenden Befehle sind als Batch-Dateien im Verzeichnis `$RVS_HOME\bin` vorhanden (Siehe bitte Kapitel 1.6 für eine detaillierte Erklärung von `$RVS_HOME`), so dass Sie sich in diesem Verzeichnis befinden müssen, um diese Programme ausführen zu können.

Schritte

- Der erste Schritt ist immer, die Konfiguration der zu administrierenden Station, RCI, anzufordern. Für diesen Zweck dient das Programm `orderConfiguration`.
**Beispiel**: Wenn Sie die Konfiguration der rvsEVO Station `TINY11` ändern möchten, müssen Sie in der Eingabeaufforderung (Kommandozeile) das Programm `orderConfiguration` mit dem folgenden Befehl starten:

```
orderConfiguration -s TINY11
```

Dieser Befehl erzeugt die Konfigurationsanforderungsdatei `cfg.req.jar` (Siehe Kapitel 13.1 für die Erklärung der `cfg.req.jar`) und sendet diese Datei via OFTP zur Station `TINY11`. Die Übertragung der Datei `cfg.req.jar` können Sie in der rvsEVO GUI (Admin - Fenster) verfolgen. Mit dem Empfang der Datei `cfg.req.jar` bei der RCI (`TINY11`), wird ein Konfigurationsprozess angestoßen. Dieser Prozess stoppt rvsEVO (bei der Station `TINY11`), archiviert die aktuelle Konfiguration in der Datei `cfg.rsp.jar`, startet rvsEVO (`TINY11`) wieder und sendet die Konfigurationsantwortdatei `cfg.rsp.jar` zurück zur LCI (rvsEVO-Station des Administrators). Im Falle, dass dieser Schritt erfolgreich war, bekommen Sie in der Eingabeaufforderung die Meldung „OrderConfiguration exited with return code *0*“. Alle diese Schritte führt das Programm `orderConfiguration` automatisch aus.

Der nächste Schritt ist das Kopieren der RCI-Konfiguration, die als Datei `cfg.rsp.jar` empfangen wurde. Das ist die Aufgabe des Programms `prepareUpdateStation`. Dieses Programm kopiert für Sie die empfangenen Konfigurationsantwortdatei `cfg.rsp.jar` und speichert diese ins WorkDir - Verzeichnis (Siehe Kapitel 13.2 für die Erklärung von WorkDir).

**Beispiel:**

```
prepareUpdateStation -s TINY11
```

**Ergebnis:** Das Verzeichnis `C:\rvsEVO\management\mgmt-workspace\TINY11\UPDATE_STATION_040828_113315\out` mit der kompleten Konfiguration der Station `TINY11` wird erzeugt. Die Meldung über den Erfolg oder Misserfolg dieses Schrittes finden Sie in der Datei `activity.log`. Diese log - Datei befindet sich im Verzeichnis `$RVS_HOME/management/mgmt-log`.

- Jetzt können Sie die Konfiguration der RCI (rvsEVO-Station `TINY11` im Beispiel) ändern. Sie können z.B. einen Lizenzschlüssel austauschen, die XML-Konfigurationsdatei für Stationen oder Jobstarts editieren oder durch das Ersetzen der ensprechenden .jar-Datei ein Update von rvsEVO durchführen. Mehr Details über derartige Administrationsaufgaben erhalten Sie in den Kapiteln 13.3.1, 13.3.2 und 13.3.3).
- Senden Sie die modifizierte Konfiguration zur RCI (`TINY11`). Sie müssen das ganze Verzeichnis `C:\rvsEVO\management\mgmt-workspace\TINY11\UPDATE_STATION_040828_113315)` mit dem folgenden Befehl senden:

```
commitUpdateStation -s TINY11 -d
C:\rvsEVO\management\mgmt-
workspace\TINY11\UPDATE_STATION_040828_113315
```

Dieser Befehl wird das modifizierte Verzeichnis zusammenfassen und zur RCI (`TINY11`) unter dem Dateinamen `cfg.req.jar` zurücksenden.

Nachdem die RCI-Station (`TINY11`) die Datei `cfg.req.jar` erhalten hat, wird rvsEVO gestoppt (alle diese Schritte werden vom Programm `commitUpdateStation` ausgeführt, Sie brauchen keine zusätzlichen Aktionen zu starten) und die geänderte Konfiguration wird aktualisiert. Der Job prüft zusätzlich, ob alles fehlerfrei durchgeführt wurde und sendet die Datei `cfg.rsp.jar` als Antwort zurück. Das Ergebnis dieser Aktion wird auch in der Datei `activity.log` protokolliert.

**Hinweis**: Im Falle eines Misserfolgs wird die alte Konfiguration wieder aktiviert.

### 13.3.1 Wie tausche ich einen Lizenzschlüssel aus?

Lizenzschlüssel

Dieses Kapitel beschreibt einen typischen Fall bei der Administration von rvsEVO-Stationen, nämlich wie man einen nicht mehr gültigen Lizenzschlüssel austauscht. Im nächsten Beispiel wird die Station `TINY01` die Station `TINY02` administrieren.

Voraussetzungen: Die Station `TINY01` muss die Station `TINY02` in der Stationstabelle als eine Nachbarstation haben (Siehe bitte Kapitel 3.2.1 für die Erklärung, wie man Stationen anlegt) und die Station `TINY02` muss die Station `TINY01` als als eine Nachbarstation in der Stationsliste haben.

Schritte

- Der erste Schritt, um einen Lizenzschlüssel auszutauschen, ist mit dem folgenden Befehl in der Eingabeaufforderung die Konfiguration von `TINY02` anzufordern:

  ```
  orderConfiguration -s TINY02
  ```

  Bei einem Erfolg dieser Aktion, bekommen Sie in der Eingabeaufforderung die Meldung „OrderConfiguration exited with return code *0*“ und anschließend werden Sie als Antwort auf Ihre Konfigurationsanforderung die Datei `cfg.rsp.jar` erhalten. (Das können Sie bei den **Beendeten Übertragungen** im Admin-Fenster von der `TINY01` GUI verfolgen.
- Der nächste Schritt ist, eine Kopie der `TINY02`-Konfiguration zu erzeugen (diese Konfiguration ist in der Datei `cfg.rsp.jar` enthalten). Dazu wird der folgende Befehl benutzt:

  ```
  prepareUpdateStation -s TINY02
  ```

  **Ergebnis:** Das Verzeichnis `C:\rvsEVO\management\mgmt-workspace\TINY02\UPDATE_STATION_040828_113315\out` mit der vollständigen Konfiguration der Station `TINY02` wurde erzeugt. Das Ergebnis dieser Aktion wird in der Datei `activity.log` protokolliert. Diese log - Datei befindet sich im Verzeichnis `$RVS_HOME/management/mgmt-log`.
- Jetzt können Sie den alten Lizenzschlüssel aus dem Verzeichnis `C:\rvsEVO\management\mgmt-workspace\TINY02\UPDATE_STATION_040828_113315\out\conf` nach `licenseOLD.properties` umbenennen und den neuen Lizenzschlüssel `license.properties` ins Verzeichnis `C:\rvsEVO\management\mgmt-workspace\TINY02\UPDATE_STATION_040828_113315\out\conf` kopieren. Wie Sie einen neuen Lizenzschlüssel erwerben können, lesen Sie bitte im Kapitel 2.2.
- Senden Sie die geänderte Konfiguration zur Station `TINY02` (Sie müssen das ganze Verzeichnis `C:\rvsEVO\management\mgmt-`

`workspace\TINY02\UPDATE_STATION_040828_113315)` mit dem folgenden Befehl senden:

```
commitUpdateStation -s TINY02 -d
C:\rvsTiny\management\mgmt-
workspace\TINY02\UPDATE_STATION_040828_113315
```

Dieser Befehl wird das ganze geänderte Verzeichnis zusammenfassen und unter dem Dateinnamen `cfg.req.jar` an die Station `TINY02` senden.

- Nach erfolgreichem Empfang der Datei `cfg.req.jar` bei der Station `TINY02`, wird rvsEVO von `TINY02` gestoppt (alle diese Schritte werden vom Programm `commitUpdateStation` selbständig ausgeführt); die geänderte Konfiguration wird aktualisiert; der Job prüft noch, ob alles fehlerfrei abgelaufen war und sendet die Antwort unter dem Dateinamen `cfg.rsp.jar` zurück. Das Ergebnis dieser Änderungen wird in der Datei `activity.log` protokolliert.

**Hinweis**: Im Falle eines Misserfolgs wird auf der RCI die alte Konfiguration wieder eingespielt.

### 13.3.2 Wie ändere ich die Konfiguration einer Station?

Konfiguration einer Station

Dieses Kapitel beschreibt einen typischen Fall in der Administration von rvsEVO, nämlich wie man Stationsparameter z.B. die ODETTE-ID ändern kann. Die gleiche Vorgehensweise gilt auch für andere Parameter (andere Stationsparameter, Jobstartparameter oder rvsEVO globale Parameter). In diesem Beispiel wird die Station `TINY20` die Station `TINY22` administrieren.

Voraussetzungen: Die Station `TINY20` muss die Station `TINY22` in der Stationstabelle haben (Siehe bitte Kapitel 3.2.1 für die Erklärung, wie man Stationen anlegt) und die Station `TINY22` muss die Station `TINY20` als eine Nachbarstation in der Stationstabelle haben.

Schritte

- Um die Station von `TINY22` entfernt konfigurieren zu können, muss die Station `TINY20` zuerst die Konfiguration von `TINY22` anfordern. Das geschieht mit dem folgenden Befehl:

  ```
  orderConfiguration -s TINY22
  ```

  Bei einem Erfolg dieser Aktion bekommen Sie in der Eingabeaufforderung die Meldung „OrderConfiguration exited with return code *0*“ und anschließend werden Sie als Antwort auf Ihre Konfigurationsanforderung die Datei `cfg.rsp.jar` erhalten. (Das können Sie bei den **Beendeten Übertragungen** im Admin - Fenster von der `TINY20` GUI verfolgen.

- Der nächste Schritt ist, eine Kopie der `TINY22` - Konfiguration im WorkDir - Verzeichnis von `TINY20` zu erzeugen (diese Konfiguration ist in der Datei `cfg.rsp.jar` enthalten). Dazu wird der folgende Befehl benutzt:

```
prepareUpdateStation -s TINY22
```

**Ergebnis:** Das Verzeichnis `C:\rvsEVO\management\mgmt-workspace\TINY22\UPDATE_STATION_040828_113315\out` mit der vollständigen Konfiguration der Station `TINY22` wurde erzeugt. Das Ergebnis dieser Aktion wird in der Datei `activity.log` protokolliert. Diese log - Datei befindet sich im Verzeichnis `$RVS_HOME/management/mgmt-log`.

- Jetzt können Sie die Datei `rvsStationlist.xml` aus dem Verzeichnis `C:\rvsEVO\management\mgmt-workspace\TINY22\UPDATE_STATION_040829_133315\out\conf` editieren und z.B. den Parameter ODETTE_ID oder TCP/IP - Adresse (Parameter IP_ADDR) für die lokale Station von `TINY22` (STATION_LOC) oder für andere Stationen (Nachbarstationen oder geroutete Stationen) ändern.
- Der nächste Schritt ist, die geänderte `TINY22`-Konfiguration an diese Station zu senden. Sie müssen das ganze Verzeichnis (aber ohne Unterverzeichnis `out`) `C:\rvsEVO\management\mgmt-workspace\TINY02\UPDATE_STATION_040829_133315)` mit dem folgenden Befehl senden:

  ```
  commitUpdateStation -s TINY22 -d
  C:\rvsEVO\management\mgmt-
  workspace\TINY22\UPDATE_STATION_040829_133315
  ```

  Dieser Befehl wird das ganze modifizierte Verzeichnis zusammenfassen und unter dem Dateinamen `cfg.req.jar` an die Station `TINY22` senden.
- Nach dem erfolgreichen Empfang der Datei `cfg.req.jar` bei der Station `TINY22`, wird rvsEVO auf `TINY22` gestoppt (alle diese Schritte werden vom Programm `commitUpdateStation` selbständig ausgeführt); die geänderte Konfiguration wird aktualisiert; der Job prüft noch, ob alles fehlerfrei abgelaufen war und sendet die Antwort unter dem Dateinamen `cfg.rsp.jar` zurück. Das Ergebnis dieser Änderungen wird auf der LCI in der Datei `activity.log` protokolliert.

**Hinweis**: Im Falle eines Misserfolgs wird auf der RCI die alte Konfiguration wieder eingespielt.

### 13.3.3 Wie führe ich ein Update von rvsEVO durch?

Update

Dieses Kapitel beschreibt einen typischen Fall in der Administration von rvsEVO-Stationen, nämlich wie man ein Update einer anderen rvsEVO - Station durchführt. In diesem Beispiel wird die Station `TINY30` die Station `TINY33` administrieren.

Voraussetzungen: Die Station `TINY30` muss die Station `TINY33` in der Stationstabelle haben (Siehe bitte Kapitel 3.2.1 für die Erklärung, wie

man Stationen anlegt) und die Station TINY33 muss die Station TINY30 in ihrer Stationstabelle haben.

Schritte

- Um die Station TINY33 updaten zu können, muss die Station TINY30 zuerst die Konfiguration von TINY33 anfordern. Das geschieht mit dem folgenden Befehl:

  `orderConfiguration -s TINY33`

  Bei einem Erfolg dieser Aktion bekommen Sie in der Eingabeaufforderung die Meldung „OrderConfiguration exited with return code *0*“ und anschließend werden Sie als Antwort auf Ihre Konfigurationsanforderung die Datei cfg.rsp.jar erhalten. (Das können Sie bei den **Beendeten Übertragungen** im Admin - Fenster von der TINY30 GUI verfolgen.

- Der nächste Schritt ist, eine Kopie der TINY33 - Konfiguration im WorkDir - Verzeichnis von TINY30 zu erzeugen (diese Konfiguration ist in der Datei cfg.rsp.jar enthalten). Dazu wird der folgende Befehl benutzt:

  `prepareUpdateStation -s TINY33`

  **Ergebnis:** Das Verzeichnis C:\rvsEVO\management\mgmt-workspace\TINY33\UPDATE_STATION_040830_113315\out mit der vollständigen Konfiguration der Station TINY33 wurde erzeugt. Das Ergebnis dieser Aktion wird in der Datei activity.log protokolliert. Diese log - Datei befindet sich im Verzeichnis $RVS_HOME/management/mgmt-log.

- Benennen Sie jetzt die alte .jar-Datei rvs.jar aus dem Verzeichnis C:\rvsEVO\management\mgmt-workspace\TINY33\UPDATE_STATION_040830_113315\out\lib nach rvsOLD.jar um und ersetzen Sie die alte .jar-Datei durch eine neue. Um eine aktuelle .jar-Datei für das Update zu bekommen, wenden Sie sich bitte an das rvs® Service Support Center (Kontakt: aus Deutschland Tel. 0800 664 77 45, aus dem Ausland Tel. +49 375 606 19 902, E-Mail: rvs-service@t-systems.com).

- Senden Sie ist die geänderte TINY33 - Konfiguration zu dieser Station. Sie müssen das ganze Verzeichnis (aber ohne Unterverzeichnis out) C:\rvsEVO\management\mgmt-workspace\TINY33\UPDATE_STATION_040829_133315) mit dem folgenden Befehl senden:

  ```
  commitUpdateStation -s TINY33 -d
  C:\rvsEVO\management\mgmt-
  workspace\TINY02\UPDATE_STATION_040830_113315
  ```

  Dieser Befehl wird das ganze modifizierte Verzeichnis zusammenfassen und unter dem Dateinamen cfg.req.jar an die Station TINY33 senden.

- Nach dem erfolgreichen Empfang der Datei cfg.req.jar bei der Station TINY33, wird rvsEVO auf TINY33 gestoppt (alle diese

Schritte werden vom Programm `commitUpdateStation` selbständig ausgeführt); die geänderte Konfiguration wird aktualisiert; der Job prüft noch, ob alles fehlerfrei abgelaufen war und sendet die Antwort unter dem Dateinamen `cfg.rsp.jar` zurück. Das Ergebnis dieser Änderungen wird auf der LCI in der Datei `activity.log` protokolliert.

**Hinweis**: Im Falle eines Misserfolgs wird auf der RCI die alte Konfiguration wieder eingespielt.

# 14 Die Kommandozeilenschnittstelle `rvsbat`

rvsbat

Die Kommandozeilenschnittstelle `rvsbat` dient der Kompatibilität mit `rvs® portable`. Mit `rvsbat` bietet `rvsEVO` eine weitere Möglichkeit zum automatisierten Versenden von Dateien und zum Verwalten von Jobstarts.

Folgende rvsbat-Kommandos werden von `rvsEVO` unterstützt:

– `SEND`: Erstellen von Sendeaufträgen

– `RESENTR`: Erstellen, Überarbeiten und Löschen von Jobstarts nach Empfang. (Entspricht den residenten Empfangseinträgen in `rvs® portable`)

– `SENDJOB`: Erstellen, Überarbeiten und Löschen von Jobstarts nach Sendeversuch.

– `FAILURE`: Erstellen, Überarbeiten und Löschen von Jobstarts nach einem Jobverarbeitungsfehler.

## 14.1 `rvsbat` starten

`rvsbat` kann wie folgt über die Kommandozeile aufgerufen werden:

**Syntax:**

```
rvsbat [/c] [/i<input file>] [/l<language>] [/q]
```

**Kommandozeilen-Parameter:**

– **/i <Inputdatei>**: Kommandos werden aus der Kommando-Eingabedatei (Input-Datei) gelesen. Die Kommando-Eingabedatei enthält folgende Elemente:
  – Kommandos (darf mehrere Zeilen lang sein; am Ende jeder fortzusetzenden Zeile muss das **+** stehen)
  – Kommentarzeilen (beginnend mit *****)

– **/c** : nach einem Fehler während der Ausführung des Kommandos `rvsbat` fortsetzten. Im Standard beendet `rvsbat` nach einem Fehler die Ausführung.

– **/q** : Kommando ohne Echo (Stumm-Modus) ausführen; Rückmeldungen über Erfolg oder Fehler der Operation werden weiterhin unterstützt.

– **/l <language>:** Nachrichtensprache verwenden, wie in **language** angegeben. Mögliche Werte: **D** (deutsch), **E** (englisch)

Die `rvsbat`-Kommandos (`SEND, RESENTR, SENDJOB und FAILURE`) können über eine Input-Datei eingelesen oder direkt in das Eingabeaufforderungsfenster geschrieben werden. Im zweiten Fall kann es mit **<STRG> C** abgebrochen werden. Die Syntax des Kommandos ist in beiden Fällen identisch.

**Beispiel:**

```
SEND /C SIDORIG=LOC DSN=C:\docs\test.txt (SID=EVO54E
DSNNEW=TESTVDSN)
```

**Kommando über Input-Datei einlesen:**

Im Folgenden wird am Beispiel des Kommandos **`SEND`** beschrieben, wie `rvsbat`-Befehle über eine Input-Datei eingelesen werden:

- Versanddatei bereitstellen (Beispiel: `test.txt`)
- Jobdatei mit `rvsbat`-Befehlen bereitstellen (enthält Versandparameter wie Ziel, OFTP-Namen, ...). Eine Jobdatei (Beispiel: `input.txt`) könnte folgendermaßen aussehen:

  **Beispiel:**

  ```
  SEND /C SIDORIG=LOC DSN=C:\docs\test.txt (SID=EVO54E
  DSNNEW=TESTVDSN)
  ```

  In diesem Beispiel wird ein Sendeauftrag erzeugt, um die Datei `C:\docs\test.txt` von der lokalen Station `LOC` an die Station `EVO54E` unter dem Namen `TESTVDSN` zu versenden.
- Das `rvsbat`-Kommando mit der Jobdatei `input.txt` als Eingabe aufrufen (im Beispiel mit Nachrichtensprache deutsch).

  **Beispiel:**

  ```
  rvsbat /LD /Ic:\docs\input.txt
  ```

**Kommando über Eingabeaufforderungsfenster aufrufen:**

Wenn Sie die rvsbat-Befehle direkt in das Eingabeaufforderungsfenster schreiben möchten, gehen Sie bitte wie im folgenden Beispiel - anhand des Kommandos `SEND` beschreiben - vor:

- `rvsbat` über das Eingabeaufforderungsfenster aufrufen (im Beispiel mit Nachrichtensprache deutsch).

  **Beispiel:**

  ```
  rvsbat /LD
  ```
- Nach erfolgreichem Start von `rvsbat` (Meldung „Kommando ’`START /USER`’ erfolgreich ausgefuehrt“ wurde ausgegeben) die `rvsbat`-Kommandos in das Eingabeaufforderungsfenster schreiben.

  **Beispiel:**

  ```
  SEND /C SIDORIG=LOC DSN=C:\docs\test.txt (SID=EVO54E
  DSNNEW=TESTVDSN)
  ```

Ein erfolgreicher Versand wird in beiden Fällen durch die Meldung „Kommando ’`SEND /CREATE`’ erfolgreich ausgeführt“ bestätigt.

Eine genaue Beschreibung der Kommandos finden Sie in den folgenden Abschnitten.

## 14.2 Sendeauftrag mit dem Kommando `SEND` erstellen.

Als Ergänzung zur GUI und zum Kommandozeilentool `createSendJob` bietet `rvsEVO` die Möglichkeit, über die Kommandozeilenschnittstelle `rvsbat` mit dem Kommando `SEND /CREATE` (auch `SEND /C`) einen Sendeauftrag zu erstellen.

**Von rvsEVO unterstützte `Send /Create` Parameter:**

Erforderlliche Parameter:

| | |
|---|---|
| **DSN** | Name und Pfad der zu sendenden lokalen Datei. (GUI-Bezeichnung: **Datei)** |
| **SID** | Stations-ID des Empfängers. Dieser Parameter ist in Klammern einzugeben. (GUI-Bezeichnung: **SID Empfänger**) |

Optionale Parameter:

| | |
|---|---|
| **CODEIN** | Code der lokalen Datei (**A**=ASCII, **E**=EBCDIC)<br>Standard: lokaler Code des Systems |
| **DISP** | Disposition für die lokale Datei, nachdem der Sendeauftrag erfolgreich ausgeführt ist. Mögliche Werte:<br>– **K**=keep, halten (default)<br>– **D**=löschen (delete).<br>(GUI-Bezeichnung: **Verbleib**) |
| **FORMAT** | das für die ODETTE-Übertragung benutzte Format. Mögliche Werte:<br>– **T**=Text (eine Folge von ASCII-Zeichen),<br>– **U**= unstrukturiert (binär),<br>– **F**=feste Satzlänge,<br>– **V**=variable Satzlänge;<br>Standard: **U** |
| **INITTIME** | Zeitpunkt zu dem der Sendeauftrag ausgeführt werden soll. Mögliche Werte:<br>– Datum und Uhrzeit im Format YYYY/MM/DD HH:mm:ss<br>– **H** bzw. **HOLD**: anhalten<br>– **N** bzw. **NOW**: sofort versenden (standard)<br>(GUI-Bezeichnung: **Ausführungstermin**) |
| **LABEL** | Benutzerkennung (bis zu 20 Zeichen) für die Serialisierung eines vorangehenden Sendeauftrages (bei **SERIAL=Y**). Alle Dateien, die in der gleichen Gruppe versendet werden, müssen die gleiche Kennung (**LABEL**) haben |
| **MAXRECL** | maximale Satzlänge für Dateien im Format **F** oder **V. (**GUI-Bezeichnung: **Max.Satzlänge**) |

| | |
|---|---|
| **SERIAL** | Diese Option gewährleistet, dass der Sendeauftrag in der von Ihnen eingegebenen Reihenfolge abgearbeitet wird (siehe auch **LABEL**). Die Datei wird erst gesendet, wenn der vorhergehende Sendeauftrag vollständig erledigt ist. Mögliche Werte:<br>– **Y** = Ja<br>– **N** = Nein<br>(GUI-Bezeichnung: **Serialisierung**) |
| **TSTAMP** | keine Funktion; dient der Kompatibilität mit `rvs® portable` Mögliche Werte:<br>– **Y** / **N** (Standard) |
| **VFTYP** | In diesem Parameter können Sie für Dateien im Format 'F' oder 'V' angeben, ob es sich um eine Textdatei oder um eine Binärdatei handelt. Mögliche Werte:<br>– **X**: Textdatei (entspricht bei rvsEVO den Satzmodus ’**TXT**’)<br>– **U**: Binäre Datei (entspricht bei rvsEVO dem Satzmodus ’**BIN**’)<br>Für weitere Informationen sehen Sie bitte die Beschreibung des Sendeparametes **Satzmodus** im Kapitel 4.5. (GUI-Bezeichnung: **Satzmodus**) |

Optionale, in Klammern einzugebende Parameter

| | |
|---|---|
| **ALG** | Dieser Parameter steht nur dann zur Verfügung, wenn **sfs = 4** / OFTP 2.0 (CMS).<br>Folgende Algorithmen stehen zur Auswahl:<br>– Kein<br>– 3DES)<br>– AES<br>(GUI-Bezeichnung: **Verschlüsselungs-algorhitmus**) |
| **CODEOUT** | Gewünschter Code (**A**=ASCII, **E**=EBCDIC) der Datei beim Empfänger. Z.B.<br>`send /c dsn=C:\test22.dat CODEIN=A FORMAT=V(SID=RTT CODEOUT=E DSNNEW=FIX0GBE.TEXT)` |

| | |
|---|---|
| **CODETABLE** | Definiert die Codetabelle, die zur Codeumwandlung verwendet werden soll (siehe Parameter **Konvertierungstabelle** in der Tabelle “Sendeparameter” auf Seite 103). Es ist der Alias-Name der Konvertierungstabelle anzugeben.<br>Beispiel: `send /c dsn=c:\programs\rvsEVO\files\outbox\test22.dat FORMAT=V(SID=RTT CODETABLE=ASCII-IBM037 MAXRECL=80 DNSNEW=FIX0GBE.TEXT)`. (GUI-Bezeichnung: **Konvertierungstabelle**) |
| **COMPRESSION** | Komprimierung während der Übertragung; nur möglich mit **sfs=2, 3 oder 4**; Mögliche Werte:<br>– **Y** (Yes)<br>– **N** (No/Standard)<br>Der Parameter muss beim Aufruf nach dem **SID** Parameter in Klammern angegeben werden; z.B. `SEND /C DSN=\home\test\test11.txt (SID=RTZ COMPRESSION=Y)`.<br>(GUI-Bezeichnung: **Offline Komprimierung**) |
| **DSNNEW** | Für die Übertragung benutzter virtueller Dateiname (**VDSN**).<br>Beim Senden an einen MVS Host-Rechner: geben Sie unbedingt einen **VDSN** an, der den MVS Namensvereinbarungen entspricht und mit einem High-Level-Begrenzungszeichen beginnt, das zu RACF passt. (GUI-Bezeichnung: **VDSN**) |
| **ENCRYPTION** | Dateiverschlüsselung; nur möglich mit **sfs=2, 3 oder 4**; Mögliche Werte:<br>– **Y** (Yes) Datei wird vor dem Senden verschlüsselt.<br>– **N** (No) keine Verschlüsselung<br>Der Parameter muss beim Aufruf nach dem **SID** Parameter in Klammern angegeben werden; z.B. `SEND /C DSN=\home\test\test11.txt (SID=RTZ COMPRESSION=Y ENCRYPTION=Y)`.<br>(GUI-Bezeichnung: **Verschlüsselung**) |
| **FILEDESC** | Dateibeschreibung, möglich nur mit **sfs=4** (OFTP 2.0)<br>(GUI-Bezeichnung: **File description)** |
| **SFS** | Sicherheitsmerkmale (**S**ecurity **F**eature **S**et); Angabe des Verschlüsselungformats. Mögliche Werte:<br>– **1** (keine Verschlüsselung; Standard)<br>– **2** (ComSecure V1)<br>– **3** (ComSecure V2<br>– **4** (OFTP 2.0)<br>(GUI-Bezeichnung: **Sicherheitsmerkmale)** |

| | |
|---|---|
| **SIDORIG** | StationsID des Absenders: lokale Station (Standard) oder virtuelle Station (GUI-Bezeichnung: **SID Absender**) |
| **SIGN** | Dateisignatur wird aktiviert. Nur mit **sfs=4** (OFTP 2.0). |
| **SIGNRESP** | Signierter EERP/NERP (Quittungssignatur) wird beantragt. Nur mit **sfs=4** (OFTP 2.0). |
| **XID** | Parameter für die Angabe einer externen JobID. Eine externe JobID kann sich auf mehrere rvsEVO-JobIDs beziehen.Sie ist freiwählbar und besteht aus alphanumerischen Zeichen. |

## 14.3 Verwaltung von Jobstarts mit rvsbat

Dieses Kapitel liefert eine Übersicht über die Verwaltung von Jobstarts mittels `rvsbat`. Es können Jobstarts nach Empfang (Kommando `RESENTR`), nach Sendeversuchen (Kommando `SENDJOB`) und nach Scheitern (Kommando `FAILURE`) erstellt, überarbeitet oder gelöscht werden. Sehen Sie bitte auch Kapitel 3.3 "Konfiguration der Jobstarts", um weitere Informationen über Jobstarts zu erhalten.

### 14.3.1 Das Kommando RESENTR

Mit dem Kommando `RESENTR` bietet `rvsbat` eine weitere Möglichkeit einen Jobstart nach Empfang zu erstellen, zu ändern oder zu löschen (entspricht dem residenten Empfangseintrag in `rvs® portable`).

**Bestimmungsworte:**

– `/CREATE` oder `/C`: erstellt einen Jobstart nach Empfang
– `/UPDATE` oder `/U`: überabeitet einen Jobstart nach Empfang
– `/DELETE` oder `/D`: löscht einen Jobstart nach Empfang

**Hinweis**: Die Bestimmungsworte können in Lang- oder Kurzform und mit oder ohne Leerzeichen an ein Kommando angehängt werden. (z.B. ist `RESENTR /DELETE DSN="*.TENNIS.*" SID="*"` gleich `RESENTR/D DSN="*.TENNIS.*" SID="*"`)

**Von rvsEVO unterstützte RESENTR Parameter:**

Erforderliche Parameter: DSN, SID

Optionale Parameter: CODETABLE, CODETRANS, COMMENT, DSN, ENABLED, EXECSYNC, EXECTIMEOUT, JOB, NEWDIR, NEWNAME, PARAMHANDLING, REPLACE, SHELL, SIDDEST, TSTAMP, TSTAMPFORMAT, VFTYP

Parameter ohne Funktion (dienen der Kompatibilität mit `rvs® portable`): ACCOUNT, DISP, FLAGCOMP, FLAGCRYPT, LUID, UID

Eine Beschreibung der Parameter finden Sie in Kapitel 14.3.4

**1. Beispiel**:

```
RESENTR/CREATE LUID="*" DSN="*.TENNIS.*" UID="*" SID="*"
SIDORIG="*" DSNNEW="" REPLACE="N" DISP="K" TSTAMP="N"
JOB="C:\rvsTest\scripts\resentr.bat"
```

In diesem Beispiel wird nach Empfang von Dateien, deren virtueller Dateiname den Zeichensatz `".TENNIS."` beinhaltet, der Job `"C:\rvsTest\scripts\resentr.bat"` gestartet, gleichnamige Dateien werden nicht überschrieben, ein Zeitstempel wird nur bei Bedarf angeängt. Die Parameter **LUID**, **UID** und **DISP** haben keine Funktion und wurden lediglich aus Kompatibilitätsgründen mit `rvs® portable` angegeben.

**2. Beispiel**:

```
RESENTR /D DSN="Test.txt" SID="*"
```

In diesem Beispiel werden alle Jobstarts gelöscht, die beim Empfang der Datei `Test.txt` gestartet werden sollen.

### 14.3.2 Das Kommando SENDJOB

Mit dem Kommando `SENDJOB` bietet `rvsbat` eine zusätzliche Möglichkeit, einen Jobstart nach Sendeversuch zu erstellen, zu ändern oder zu löschen.

**Bestimmungsworte:**

– `/CREATE` oder `/C`: erstellt einen Jobstart nach Sendeversuch
– `/UPDATE` oder `/U`: überabeitet einen Jobstart nach Sendeversuch
– `/DELETE` oder `/D`: löscht einen Jobstart nach Sendeversuch

**Von rvsEVO unterstützte SENDJOB Parameter:**

Erforderlliche Parameter: SID (= SIDDEST), VDSN

Optionale Parameter: ATTEMPTS, COMMENT, ENABLED, EXECSYNC, EXECTIMEOUT, JOB, PARAMHANDLING, SHELL, SIDSENDER = SIDORIG, UID

Eine Beschreibung der Parameter finden Sie im Kapitel 14.3.4.

**1. Beispiel:**

```
SENDJOB /C VDSN="Test.txt" SID="RVS" ATTEMPTS=0
SIDSENDER="LOC" JOB="C:\rvsTest\scripts\sendjob.bat"
CODETRANS="" CODETABLE="" FLAGCOMP="N" FLAGCRYP="N"
```

In diesem Beispiel wird nach erfolgreichem Versand (**ATTEMPTS**=0) der Datei `"Test.txt"` von der Station `LOC` an die Station `RVS` der Job `"C:\rvsTest\scripts\sendjob.bat"` gestartet. Eine

Codeumwandlung findet nicht statt, die Parameter **FLAGCOMP** und **FLAGCRYP** haben keine Funktion.

**2. Beispiel:**

```
SENDJOB /D VDSN="Test.txt" SID="RVS"
```

In diesem Beispiel werden alle Jobstarts gelöscht, die beim Versand der Datei `"Test.txt"` an die Station `RVS` gestartet werden sollen.

### 14.3.3 Das Kommando FAILURE

Mit dem Kommando `FAILURE` bietet `rvsbat` eine weitere Möglichkeit, einen Jobstart nach einem Jobverarbeitungsfehler zu erstellen, zu ändern oder zu löschen.

**Bestimmungsworte:**

– `/CREATE` oder `/C`: erstellt einen Jobstart nach Jobverarbeitungsfehler
– `/UPDATE` oder `/U`: überabeitet einen Jobstart nach Jobverarbeitungsfehler
– `/DELETE` oder `/D`: löscht einen Jobstart nach Jobverarbeitungsfehler

**Unterstützte FAILURE Parameter:**

Erforderllicher Parameter: VDSN

Optionale Parameter: COMMENT, ENABLED, EXECSYNC, EXECTIMEOUT, JOB, PARAMHANDLING, SHELL, SIDDEST, SIDORIG

Eine Beschreibung der Parameter finden Sie in Kapitel 14.3.4.

**Beispiel:**

```
FAILURE/C VDSN="Test.txt" SIDORIG="LOC" SIDDEST=“RVS“
JOB="C:\rvsTest\scripts\failure.bat"
```

In diesem Beispiel wird nach Erhalt einer Fehlermeldung beim Versand der Datei `"Test.txt"` von der Station `LOC` an die Station `RVS` der Job `"C:\rvsTest\scripts\failure.bat"` gestartet

### 14.3.4 Beschreibung der Jobstart-Parameter

In der folgenden Tabelle finden Sie eine Beschreibung der Parameter aller Jobstarts:

Jobstart-Parameter

Erforderlliche Parameter:

| | |
|---|---|
| **DSN** | (nur RESENTR) Virtueller Dateiname, max. Länge 26 Zeichen. Zulässing sind der SID-Zeichensatz sowie der Platzhalter ’*****’ am Ende. (GUI-Bezeichnung: **VDSN**) |

| | |
|---|---|
| **SID**<br>oder **SIDDEST** | (nur SENDJOB) Stations-ID des Empfängers. Zulässing sind der SID-Zeichensatz sowie der Platzhalter '*****' am Ende.<br>(GUI-Bezeichnung: **SID Empfänger**) |
| **SID** | (nur RESENTR) Stations-ID des Absenders. Zulässing sind der SID-Zeichensatz sowie der Platzhalter '*****' am Ende.<br>(GUI-Bezeichnung: **SID Absender**). |
| **VDSN** | (SENDJOB und FAILURE) Name und Pfad der zu sendenden lokalen Datei. Zulässing sind der SID-Zeichensatz sowie der Platzhalter '*****' am Ende. |

Optionale Parameter:

| | |
|---|---|
| **ACCOUNT** | keine Funktion; dient der Kompatibilität mit `rvs® portable` |
| **ATTEMPTS** | Anzahl der fehlgeschlagenen Sendeversuche. Wird hier "0" eingetragen, so bedeutet dies erfolgreicher Dateiversand.<br>(GUI-Bezeichnung: **Sendeversuche**) |
| **CODETABLE** | Definiert die Codetabelle, die zur Codeumwandlung verwendet werden soll, wenn Parameter **CODETRANS** = '**T**'. Es ist der Alias-Name der Konvertierungstabelle anzugeben (siehe Parameter **Conversion Table** in der Tabelle "Unterelemente von Jobfilter" auf Seite 86).<br>Beispiel: `RESENTR /C DSN="C:\files\test22.dat" CODETRANS="T" CODETABLE="ASCII-IBM037" DSNNEW="TEST22"` |
| **CODETRANS** | Dieser Parameter gibt an, ob eine Code-umwandlung stattfinden soll. Mögliche Werte:<br>– Keine Angabe: es findet keine Codeumwandlung statt<br>– **E**: EBCDIC - ASCII Umwandlung<br>– **A**: ASCII - EBCDIC Umwandlung<br>– **T**: eingene, in Parameter **CODETABLE** angegebene Tabelle, wird verwendet. |
| **COMMENT** | beliebiger Text;<br>GUI-Bezeichnung: **Kommentar**; |
| **DISP** | keine Funktion; dient der Kompatibilität mit `rvs® portable` Wert: **K** |

| | |
|---|---|
| **DSNNEW** | Mittels dieses Parameters können Sie einer empfangenen Datei einen neuen Pfad/Namen Namen zuweisen. Sie haben folgende Möglichkeiten:<br>– keine Angabe: Datei wird mit aktuellem Namen in Inbox abgelegt<br>– Angabe des absoluten Pfads und Datei-namens: Zustellung in angegebenes Verzeichnis mit neuem Namen |
| **ENABLED** | Der Parameter legt fest, ob der Job gestartet werden soll. Mögliche Werte:<br>– **Y** (Yes): Job starten (Standard)<br>– **N** (No): Job nicht starten<br>(GUI-Bezeichnung: **Aktiviert**) |
| **EXECSYNC** | Dieser Paramter gibt an, ob die OFTP-Session wird so lange aufrecht erhalten werden soll, bis das Programm abgearbeitet wurde. Mögliche Werte:<br>– **Y** = Ja<br>– **N** = Nein (Standard)<br>(GUI-Bezeichnung: **Synchronisiert**) |
| **EXECTIMEOUT** | Bei Einstellung **EXECSYNC** = **Y**: Zeit nach der die Verbindung abgebrochen wird.<br>(GUI-Bezeichnung: **Timeout (Sync.)**) |
| **FLAGCOMP** | keine Funktion; dient der Kompatibilität mit `rvs® portable`; Mögliche Werte: **N** / leerer String |
| **FLAGCRYPT** | keine Funktion; dient der Kompatibilität mit `rvs® portable`; Mögliche Werte: **N** / leerer String |
| **JOB** | Programm, das gestartet werden soll, wenn die Filterbedingungen alle zutreffen. Den Programmen wird ein fest definierter Satz von Parametern übergeben.<br>(GUI-Bezeichnung: **Prozess**) |
| **LUID** | keine Funktion; dient der Kompatibilität mit `rvs® portable`; Mögliche Werte: ***** / leerer String |
| **NEWDIR** | Mittels dieses Parameters können Sie einer empfangenen Datei einen neuen Pfad zuweisen. Sie haben folgende Möglichkeiten:<br>– keine Angabe: Zustellung in Inbox<br>– Angabe des absoluten Pfads: Zustellung in angegebenes Verzeichnis (anstelle von Inbox)<br>(GUI-Bezeichnung: **Neues Verzeichnis**) |

| | |
|---|---|
| **NEWNAME** | Mittels dieses Parameters können Sie einer empfangenen Datei einen neuen Namen zuweisen. Sie haben folgende Möglichkeiten:<br>– keine Angabe: VDSN wird als Dateiname genommen<br>– Angabe des Dateinamens: Zustellung mit neuem Namen<br>(GUI-Bezeichnung: **Neuer Dateiname)** |
| **PARAMHANDLING** | Mit diesem Parameter können Sie die Übergabe der Jobdaten an den Prozess einstellen. Mögliche Werte:<br>– **ARGS**: Jobdaten werden als Parameter / Argumente an das Skript übergeben<br>– **ENV**: Jobdaten werden als Umgebungsvariablen gesetzt.<br>– **REPLACE**: Dient der Kompatibilität mit rvs® portable (Standard).<br>Für weitere Informationen lesen Sie bitte den Abschnitt “Parameterübergabe” auf Seite 90.<br>GUI-Bezeichnung: **Parameterübergabe**) |
| **REPLACE** | Mit Replace legen Sie fest, wie mit einer empfangenen Datei verfahren wird, die im Namen mit einer lokalen Datei übereinstimmt. Mögliche Werte:<br>– **R** oder **Y** (yes): vorhandene Datei wird überschrieben<br>– **N** (no): Datei wird nicht überschrieben, Zeitstempel wird an die neue Datei angehängt (Standard)<br>– **I** = ERR ???<br>(GUI-Bezeichnung: **Ersetzen**) |
| **SHELL** | Nur für UNIX-Plattformen, wenn notwendig: Kommando-Shell, in der das Programm ausgeführt werden soll; z.B. ksh,.csh, .... |
| **SIDDEST** | Stations-ID der Zielstation<br>(GUI-Bezeichnung: **SID Empfänger**) |
| **SIDORIG** | Stations-ID der Ursprungsstation (lokale oder virtuelle Station)<br>(GUI-Bezeichnung: **SID Absender**) |
| **SIDSENDER** | Alternativ für **SIDORIG** |

| | |
|---|---|
| **TSTAMP** | Mit diesem Parameter legen Sie fest, nach welchen Regeln ein Zeitstempel generiert werden soll. Mögliche Werte:<br>– **Y** = Zeitstempel wird immer angehängt<br>– **N** = Zeitstempel wird nur bei Bedarf angehängt (Standard)<br>Format des Zeitstempels ist wie im Parameter **TSTAMPFORMAT** definiert.<br>(GUI-Bezeichnung: **Zeitstempel**) |
| **TSTAMPFORMAT** | Format des an den Dateinamen anzuhängenden Zeitstempels. Mögliche Werte:<br>– **TIME**: Uhrzeit der Erstellung des Jobs und fortlaufender Zähler im Format hhmmsscсc (Standard)<br>– **DATETIME**: Datum und Uhrzeit der Erstellung des Jobs und fortlaufender Zähler im Format YYMMDDhhmmssccc.<br>– **SFID_DATETIME**: Odette-Zeitstempel der Übertragung aus der SFID; Datum, Uhrzeit und fortlaufender Zähler im Format YYYYMMDDhhmmsscccc<br>– **COUNTER**: 000000 bis 999999 (fortlaufender Zähler). Ist der Zähler (z.B. durch Löschen einer Datei) nicht fortlaufend, wird erst die Lücke gefüllt, bevor der Counter weiter hochgezählt wird.<br>(GUI-Bezeichnung: **Zeitstempelformat**) |
| **UID** | keine Funktion; dient der Kompatibilität mit `rvs® portable`; Mögliche Werte:<br>– leerer String<br>– * |
| **VFTYP** | In diesem Parameter können Sie für Dateien im Format 'F' oder 'V' angeben, ob es sich um eine Textdatei oder um eine Binärdatei handelt. Mögliche Werte:<br>– **X**: Textdatei (entspricht bei rvsEVO den Satzmodus ’**TXT**’)<br>– **U**: Binäre Datei (entspricht bei rvsEVO dem Satzmodus ’**BIN**’)<br>Für weitere Informationen sehen Sie bitte die Beschreibung des Sendeparametes **Satzmodus** im Kapitel 4.5.<br>(GUI-Bezeichnung: **Recordbehandlung**) |

# 15 Anhang

## 15.1 ODETTE-Protokoll

Der Zweck des Odette Übertragungsprotokolls (OFTP)  ist die zuverlässige Übertragung von Dateien. Das OFTP eröffnet eine Protokollsitzung mit dem OFTP der entfernten Station, die logisch über der Leitungstreiberverbindung läuft.

Nach dem Beginn der OFTP Sitzung tauschen beide Seiten ihre IDs und Passwörter aus, verhandeln über Parameter, wie die Größe des Odette Austauschpuffers, den Odette Kreditwert (die Anzahl der Puffer, die die sendende Seite ohne eine Antwort abzuwarten senden kann). Sie tauschen Informationen über den Namen, die ungefähre Größe und das Format der zu übertragenden Datei aus.

Während der Übertragung findet eine Kompression und Dekompression der Daten statt. Nach der Übertragung werden die Bytezähler auf beiden Seiten verglichen. Nach der erfolgreichen Speicherung der Datei wird eine Quittierung an die Ausgangsstation gesandt. Falls die Übertragung gestört wurde, z.B. durch Leitungsunterbrechung, verfügt das OFTP Protokoll über einen Mechanismus, der die Wiederaufnahme der Übertragung an dem Unterbrechungspunkt gewährleistet.

Durch die Funktion Richtungswechsel wird der Empfänger zum Sender und ist in der Lage, Quittungen und Dateien zu übertragen.

Für mehr Protokolleinzelheiten konsultieren Sie bitte die Veröffentlichungen der Odette und VDA Gremien: "Odette Specifications for File Transfer".

In der folgenden Tabelle wird der allgemeine, aber vereinfacht dargestellte Nachrichtenfluss in einer Odette Sitzung gezeigt. Die sendende Seite ist der Initiator, die empfangende Seite arbeitet als Reagierender.

| Initiator (Sender) | | Responder (Empfänger) |
|---|---|---|
| Verbinden | ----- Netzwerke verbinden -----> | |
| | <----- SSRM ----- | „ODETTE FTP READY“ |
| Odette ID, Passwort. Größe, Kredit | ----- SSID -----> | |
| | <----- SSID ----- | Odette ID, Passwort etc. |
| **S**tart**F**ile **ID** (Name, Größe, Format) | ----- SFID -----> | |
| | <----- SFPA ----- | **S**tart**F**ile **P**ositive **A**nswer |

| | | |
|---|---|---|
| ’n’ Datenaufzeichnungen | ----- DATA -----><br>----- DATA -----><br>----- DATA -----><br>... | |
| | <----- CREDIT ----- | senden von Kreditwert ’n’ |
| senden von ’n’ Datenpaketen | ----- DATA -----><br>----- DATA -----><br>.<br>. | |
| **E**nd**F**ile **ID** (Bytezähler) | ----- EFID -----> | |
| | <----- EFPA -----<br>oder<br><----- EFNA ----- | **E**nd**F**ile **P**ositive **A**nswer (wenn speichern erfolgreich und der Bytezähler korrekt war)/ **E**nd**F**ile **N**egative **A**nswer |
| **C**hange **D**irection | ------ CD ------> | |
| | <----- EERP -----<br>oder<br><----- NERP ----- | **E**nd-to-**E**nd **R**es**p**onse/ **N**egative **E**nd-to-End **R**es**p**onse (Quittierung) |
| | <----- ESID ----- | **E**nd-**S**ession **ID** |
| Netzwerk trennen | | Netzwerk trennen |

## K

## L



## M



## N



## O

## P



## R



## S

## T



## U



## V



## W



## X